FMS^2: Unified Flow Matching for Segmentation and Synthesis of Thin Structures

Babak Asadi (✉), Peiyang Wu, Mani Golparvar-Fard,
Viraj Shah, and Ramez Hajj

University of Illinois at Urbana-Champaign
{basadi2,peiyang7,mgolpar,vjshah3,rhajj}@illinois.edu

Abstract. Segmenting thin structures such as infrastructure cracks and anatomical vessels is hampered by topology-sensitive geometry, high annotation cost, and poor cross-domain generalization. We propose FMS^2, a unified flow-matching framework with two complementary modules. ***SegFlow*** is a 2.96M-parameter encoder–decoder segmentation model that recasts prediction as continuous image→mask transport: it learns a time-indexed velocity field with a single flow-matching regression loss and obtains the mask through deterministic ODE integration, rather than supervising only end-state logits. This trajectory-level supervision improves thin-structure continuity and sharpness over topology-aware loss baselines, without auxiliary topology heads, post-processing, or multi-term loss engineering. ***SynFlow*** is a mask-conditioned mask→image generator that produces pixel-aligned image–mask pairs by injecting mask geometry at multiple scales, emphasizing boundary bands with edge-aware gating, and expanding sparsity, width, and branching through a controllable mask generator. On five crack and vessel benchmarks, SegFlow alone outperforms CNN, Transformer, Mamba, and generative baselines, improving mean IoU from 0.511 to 0.599 (+17.2%) and reducing Betti matching error from 82.145 to 51.524 (−37.3%). With limited labels, SynFlow recovers near-full performance using 25% real annotations and improves cross-domain IoU by 0.11. Unlike classical augmentation, SynFlow provides pixel-aligned supervision with controllable structural shifts (*e.g.*, sparsity, width, branching), making it effective under domain shift. We release 10k crack and 1k vessel image–mask pairs. The project page is available at FMS^2.

Keywords: Flow Matching · Thin-Structure Segmentation · Topology-Preserving Segmentation · Mask-Conditioned Image Synthesis · Label-Efficient Learning

1 Introduction

Segmenting thin structures is critical for a wide range of applications such as infrastructure inspection and medical imaging. However, current methods still fall short of expected performance levels. Thin structures can be as narrow as one pixel, occupy less than 1% of the field of view, and still require connected predictions for

downstream metrology and planning [8, 24, 32, 42, 51]. Pixel-wise annotation for ground truth is labor-intensive and exhibits inter-annotator variability, especially along low-contrast, branching structures and ambiguous margins [3, 15, 27, 31, 47]. Models trained on one dataset frequently fail to generalize when background texture, material, illumination, or acquisition protocol changes [4, 10, 39, 40, 42]. Taken together, (i) topological fragility at one-pixel scales, (ii) costly and variable supervision, and (iii) pronounced domain shift form the core challenges that most prior studies address only in part, often focusing on a single facet.

Architectural advances have improved in-domain accuracy. Multi-scale CNNs with explicit edge or contour heads [1, 24, 25, 41], CNN–Transformer hybrids [4, 48] that fuse local detail with long-range context, and recent Mamba or SSM variants [17, 18, 33, 51] that propagate information at linear cost all contribute [6, 26, 40]. Nevertheless, architecture alone is not a consistent remedy. On common thin-structure benchmarks, a vanilla U-Net baseline often matches [10, 24, 47] and occasionally exceeds the performance of more elaborate architectures [17, 48, 51]. Two factors recur. First, limited dataset scale and severe class imbalance constrain the benefits of high-capacity attention or SSM modules and increase overfitting risk. In addition, overlap-style objectives do not enforce connectivity, so added capacity often models background texture rather than preserving continuous centerlines. Under cross-dataset evaluation, per-pixel classifiers trained with overlap objectives still show large performance drops [8, 39, 42, 47].

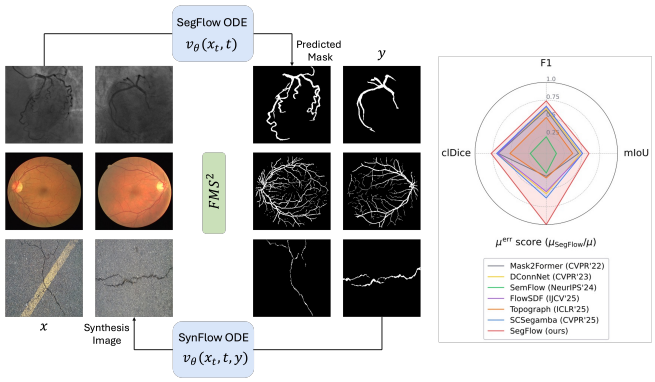

Fig. 1: FMS^2 Framework. SegFlow performs image→mask transport for topology-preserving segmentation, while SynFlow renders topology-consistent mask→image pairs to reduce label demand.

A complementary thread introduces topology-aware supervision through differentiable losses such as clDice or through graph-based post-processing that reconnects broken centerlines [1, 7, 9, 12, 20, 22, 32]. These approaches improve connectivity, but often add extra objectives, memory cost, or post-processing, and they still operate on the final prediction rather than modeling how the mask forms. A third line reduces label cost or mitigates domain shift using synthetic data from GAN or diffusion models, or through frequency and geometry alignment with unlabeled targets [26, 28, 37–40]. Generic conditioning pipelines can erase thin structures through normalization, which leads to mask–image

mismatch, and iterative diffusion adds sampling overhead. When conditioning on provided masks, topological diversity can also be constrained by the mask distribution [14, 34, 43, 45].

Our distinction from diffusion-based segmentation is not only the use of flow matching instead of denoising. In SegFlow, the observed image is the initial state of the ODE and is continuously transported into the mask, whereas diffusion segmentation typically denoises a random mask or latent variable conditioned on the image. This makes topology preservation part of the transport trajectory rather than a property recovered only at endpoint decoding. A single framework that unifies accurate thin-structure segmentation, label efficiency, and robustness to domain shift is desirable. To this end, we introduce FMS^2, a framework built on a *flow-matching* formulation [5, 16, 19, 30, 46] that systematically addresses these challenges.

Our first contribution, *SegFlow*, directly confronts the core problem of topological fragility (i). SegFlow recasts segmentation as an image→mask transport problem. Instead of predicting per-pixel logits at a single end time, it learns a time-indexed velocity field that transports an input image toward its ground-truth mask under a deterministic flow. This approach supervises the *entire formation process* of the mask, encoding topological consistency from the start of the trajectory rather than only at its final state. As our experiments show, SegFlow alone brings clear improvements in segmentation quality over established baselines.

To address the second challenge of costly supervision (ii), we propose a complementary module, *SynFlow*, a mask→image renderer designed to enhance SegFlow's label efficiency through synthetic paired data generation. SynFlow uses the same U-Net architecture, but conditions on binary masks through multi-scale spatial modulation and an edge-aware gating path. This yields pixel-aligned synthetic image–mask pairs that preserve sub-pixel boundaries and counter the normalization-induced drift seen in other generative models [28, 45].

We also introduce a controllable mask generator that produces a diverse range of masks (from ultra-sparse to dense) to be fed into SynFlow, allowing us to concentrate synthetic supervision exactly where manual annotation is most expensive and topologically complex.

Training SegFlow with synthetic pairs from SynFlow lets the segmentation model inherit topology-aware transport, while the renderer supplies diverse, topology-consistent pairs. Together, our framework confronts (i) fine-scale topology, (ii) limited supervision, and (iii) domain shift. On five crack and vessel benchmarks, SegFlow alone delivers substantial improvements in both overlap-based and topological measures over CNN, Transformer, Mamba, diffusion, and topology-aware baselines. It boosts the mean IoU from 0.511 to 0.599 (a 17.2% relative gain), raises the mean clDice from 0.697 to 0.774 (a 11.05% relative gain), and reduces the Betti matching error from 82.145 to 51.524 (−37.3%). The addition of SynFlow provides significant gains in label efficiency and robustness. With only 25% human-annotated masks, training SegFlow with SynFlow-generated pairs recovers performance close to full supervision. Furthermore, under source-only cross-dataset evaluation, augmenting the source domain with SynFlow

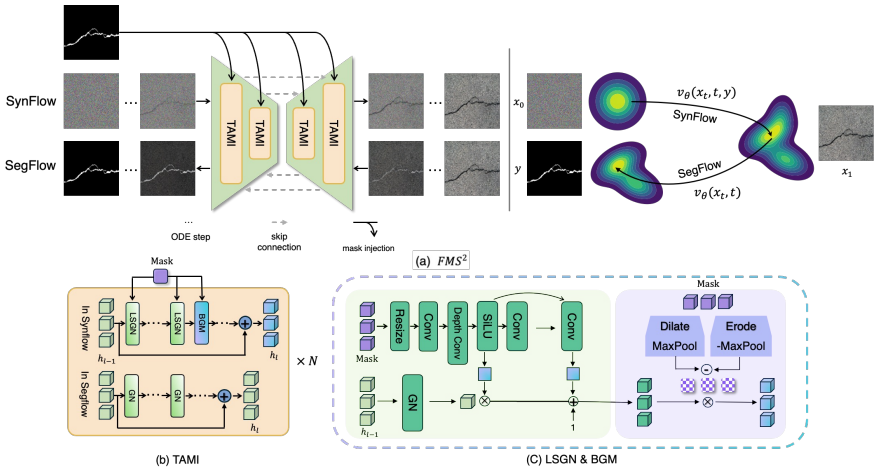

Fig. 2: Architecture and transport view of FMS^2. (a) SegFlow and SynFlow share a U-Net backbone and are trained with flow matching: SegFlow evolves the image toward the binary mask, while SynFlow performs mask-conditioned synthesis with multi-scale mask injection. **(b)** Topology-Aware Mask Injection (TAMI) uses lite-SPADE GroupNorm (LSGN) and the boundary-gating module (BGM) to inject mask geometry and focus capacity near thin structures. **(c)** LSGN modulates group-normalized features using a mask pyramid, while BGM gates features around crack/vessel interfaces.

images improves target-domain IoU by about **0.11** on average. In summary, our primary contributions include:

- ***SegFlow***, a flow-matching segmentation model that outperforms prior SOTA architectures and topology-aware loss designs for thin-structure segmentation, improving both overlap-based accuracy and topological fidelity without auxiliary heads or post-processing.
- ***SynFlow***, a mask-conditioned renderer with multi-scale topology injection and edge-aware gating that produces pixel-aligned, topology-preserving synthetic image–mask pairs. By mixing a small fraction of human-annotated data with SynFlow-generated samples, SynFlow notably reduces label demand and improves cross-dataset robustness.
- **A large-scale dataset** of **10k crack** and **1k vessel** pairs to support benchmarking and generalizable thin-structure models.

2 Related Work

2.1 Thin-structure segmentation

Segmentation of thin structures spans generic encoder–decoders and topology-aware models. U-Net and nnU-Net remain strong baselines in biomedical and infrastructure domains [10, 29], and for cracks and vessels, multi-scale CNNs add edge or contour heads to stabilize one-pixel traces [24, 41]. Transformer-based Mask2Former [6], hybrid CNN–Transformer designs such as nnWNet [48], and

vision SSMs like SCSegamba with structure-aware Mamba blocks [17] improve long-range context and efficiency, while DConnNet injects directional connectivity priors in the latent space for anatomy [42]. Yet cross-dataset studies still report sizable drops under appearance shift, suggesting that architecture alone does not guarantee topology preservation.

Topology-aware supervision makes connectivity explicit [9, 22]. clDice and its variants promote centerline overlap [1, 7, 32], Skeleton Recall Loss targets tubular connectivity with lower overhead [12], Topology-Aware Uncertainty estimates structure-wise uncertainty [8], and Topograph formulates a component-graph loss with preservation guarantees and efficient computation [20]. These approaches improve connectivity, but typically require dense labels plus extra loss terms or post-processing; they do not explicitly model image-to-mask transport.

2.2 Generative models for segmentation & synthesis

Generative models support segmentation, label efficiency, and robustness. In medical imaging, MedSegDiff-V2 shows that diffusion-based segmentation can outperform discriminative baselines in low-data regimes [40]. Diffusion-based segmentation generally denoises a random mask conditioned on an image, while UniGS/SemFlow-style models operate in latent or semantic spaces. FlowSDF [2] replaces stochastic denoising with deterministic flows using an implicit SDF target. In contrast, SegFlow keeps the trajectory in pixel space and uses the input image as the evolving state, providing trajectory-level image→mask supervision that better preserves one-pixel cracks and vessels.

Joint segmentation–generation frameworks such as SemFlow [38] are related, but learn a latent-space rectified flow that couples geometry and appearance. FMS^2 decouples these roles: SegFlow handles deterministic image→mask transport, while SynFlow renders *paired, pixel-aligned* mask→image samples with multi-scale topology injection. Diffusion models enable high-fidelity synthesis [28], and latent-diffusion methods extend them to panoptic label maps and mask inpainting [37]. Conditioning improves controllability: ControlNet adds spatial conditions to frozen diffusion [45], and later variants refine spatially adaptive normalization [14, 43]. However, pipelines such as LDM [28], ControlNet [45], and T2I-Adapter [23] condition generation through latent encoders, adapters, or auxiliary branches, which can weaken one-pixel geometry, introduce mask–image drift, and blur or disconnect filaments; related work reports similar topology washout [13]. SynFlow avoids this through pixel-space mask→image transport with boundary-aware multi-scale conditioning; Supplementary Table S6 and Fig. S10 show gains over these baselines.

Flow matching and rectified flows learn time-indexed velocity fields for deterministic transport [16, 19]. SemFlow binds segmentation and synthesis through a rectified-flow ODE [38], and Diff2Flow transfers diffusion quality to flows for faster sampling [30]. These pipelines target semantic regions or thicker structures, whereas FMS^2 targets ultra-sparse, sub-pixel topology and controllable mask-conditioned synthesis.

3 Methodology

3.1 Preliminaries

Flow Matching (FM) learns a time-indexed vector field that moves a state along a continuous path between two endpoints, optionally under conditioning. Let c denote the conditioning signal; in SynFlow, c is the binary mask, while in SegFlow the image and mask define the transport endpoints. A state $\mathbf{x}_t$ evolves according to the ODE

$$\frac{d\mathbf{x}_t}{dt} \;=\; \mathbf{u}_\theta(\mathbf{x}_t, t \mid c), \tag{1}$$

so integrating from t=0 to 1 maps a source state $\mathbf{x}_0$ to an endpoint $\mathbf{x}_1$. Instead of simulating the ODE during training, FM prescribes a reference path that links a noisy source to a noisy data endpoint and exposes a closed-form target velocity. We adopt a straight-line path with constant additive noise,

$$\mathbf{x}_t \;=\; (1{-}t)\,\mathbf{x}_0 \;+\; t\,\mathbf{x}_1 \;+\; \sigma\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon}{\sim}\mathcal{N}(\mathbf{0},\mathbf{I}), \tag{2}$$

whose time derivative is $\partial_t \mathbf{x}_t \;=\; \mathbf{x}_1{-}\mathbf{x}_0$. The linear term steers the state along the straight line between endpoints, while the additive Gaussian term keeps the path stochastic without affecting the target velocity. Training samples pairs $(\mathbf{x}_0, \mathbf{x}_1, c)$, draws a random $t \in [0,1]$, evaluates $\mathbf{x}_t$ via (2), and regresses $\mathbf{u}_\theta(\mathbf{x}_t, t \mid c)$ to this target velocity. At test time, a deterministic sampler is obtained by numerically integrating (1) from the seed. This flow-matching formulation underlies both the SegFlow and SynFlow ODE branches shown in Figure 2.

3.2 SegFlow: Segmentation as continuous transport

SegFlow treats image segmentation as a continuous transport problem: instead of classifying pixels independently, it learns a time-indexed velocity field that transports an input image into its ground-truth mask under a deterministic flow (upper branch in Figure 2). Concretely, let x_0 be an input image and x_1 its binary mask. We define a linear interpolation path from x_0 to x_1 (cf. Eq. (4)), so that for any $t \in [0,1]$ the state x_t lies between the image and mask. The target velocity along this path is $u_t^{\text{target}} \;=\; x_1 - x_0$, *i.e.*, the constant vector difference from image to mask. We train a U-Net–based neural ODE model $u_\theta(x_t, t)$ to regress this vector field by minimizing the mean-squared error:

$$\mathcal{L}_{\text{SegFlow}} \;=\; \mathbb{E}_{(x_0,x_1),\, t\sim[0,1]}\Big[\, \big\| \, u_\theta(x_t,t) - (x_1 - x_0) \big\|_2^2 \,\Big]. \tag{3}$$

In practice, we sample t uniformly, compute

$$x_t \;=\; (1-t)\,x_0 \;+\; t\,x_1, \tag{4}$$

then regress $u_\theta(x_t, t)$ to $x_1{-}x_0$; in implementation, we scale time as $\tau = 1000\,t$ for the U-Net time embedding.

Importantly, we do not predict logits or probabilities: at every timestep, the output of the network is a velocity field which receives a target supervision as per Eq. 3. By supervising the formation of the mask over $t \in [0, 1]$, SegFlow implicitly encourages topological consistency (*e.g.*, connected branches) early in the trajectory rather than only at the final mask. In contrast, traditional segmentation losses (cross-entropy or Dice) supervise only the end-state logits and often miss fine connectivity [6], even when augmented with topology-aware losses such as clDice [32]. Our velocity-based loss anchors the vector field at each t, so the ODE integrator must carry thin structures intact to reach the true mask.

This differs from diffusion-based segmentation in the state being evolved. Rather than denoising a random mask or latent code conditioned on the image, SegFlow starts from the raw image and evolves the image-space state toward the binary mask. The model therefore receives trajectory-level supervision on how cracks or vessels emerge from the observed appearance, forcing fragile branches to remain recoverable throughout integration instead of relying only on stochastic endpoint decoding.

At test time, SegFlow performs deterministic ODE integration from t=0 to 1 starting from the input image, using a fixed-step Euler solver with K steps. Given x_0, we iteratively update

$$x_{t+\Delta t} \;=\; x_t \;+\; \Delta t\, u_\theta(x_t, t), \tag{5}$$

where $\Delta t = 1/K$ and $t \in \{0, \Delta t, 2\Delta t, \ldots, 1 - \Delta t\}$, to obtain the final state $x_1 = \hat{M}$. We then threshold $\hat{M}$ to yield a binary segmentation.

When a thin branch would otherwise disconnect, the velocity field must repair it over t so that x_1 matches the ground-truth mask. In practice, SegFlow yields more complete and connected centerlines, preserving topology without auxiliary skeleton losses [32] or post-hoc graph heuristics.

Self-Conditioning Mechanism. For ultra-thin structures (e.g., cracks), we additionally use a self-conditioning variant (SegFlow-SC) in which the velocity network also receives a running estimate of the terminal mask. SegFlow-SC replaces the velocity network by $u_\theta^{\mathrm{SC}}([x_t, \tilde{x}_1], t)$, where $\tilde{x}_1$ is an estimate of the final mask at time t and $[\cdot, \cdot]$ denotes channel-wise concatenation.

Under the straight-line interpolant (Eq. (4)) and constant target velocity u_t^{target}, a predicted velocity implies an endpoint estimate

$$\hat{x}_1 \;=\; x_t + (1 - t)\, u_\theta^{\mathrm{SC}}([x_t, \tilde{x}_1], t). \tag{6}$$

During training, with probability p_{sc}, we first run the network with $\tilde{x}_1 = \mathbf{0}$ to form $\hat{x}_1$, stop-gradient it, and then run a second pass conditioned on this estimate:

$$\tilde{x}_1 \leftarrow \mathrm{stopgrad}\big(x_t + (1 - t)\, u_\theta^{\mathrm{SC}}([x_t, \mathbf{0}], t)\big)\,. \tag{7}$$

The loss remains the same MSE regression to the FM target velocity, but applied to the self-conditioned prediction.

3.3 SynFlow: Rendering as topology-aware transport

SynFlow synthesizes a photorealistic image that is layout- and connectivity-consistent with a given thin-structure mask. The generator is an ODE sampler conditioned on the mask: starting from a noise seed $\mathbf{z} \sim p_0$, the predicted image is the terminal state of the flow

$$\hat{\mathbf{x}} \;=\; \mathbf{z} \;+\; \int_0^1 \mathbf{u}_\theta(\mathbf{x}_t, t \mid \mathbf{M})\, dt, \tag{8}$$

implemented with a fixed-budget explicit ODE solver. In practice, we use the Dormand–Prince (`dopri5`) method; for exposition, a single explicit step can be written as

$$\mathbf{x}_{t+\Delta t} \;=\; \mathbf{x}_t \;+\; \Delta t\, \mathbf{u}_\theta(\mathbf{x}_t, t \mid \mathbf{M}). \tag{9}$$

For each training pair $(\mathbf{x}_1, \mathbf{M})$, SynFlow instantiates the mask-conditioned FM interpolant with $(\mathbf{x}_0, \mathbf{x}_1) = (\mathbf{z}, \mathbf{x}_1)$,

$$\mathbf{x}_t \;=\; (1-t)\,\mathbf{z} \;+\; t\,\mathbf{x}_1 \;+\; \sigma\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

which yields the closed-form target velocity $\partial_t \mathbf{x}_t \;=\; \mathbf{x}_1 - \mathbf{z}$; the training objective regresses the conditional velocity along this path,

$$\mathcal{L}_{\text{SynFlow}} \;=\; \mathbb{E}_{(\mathbf{x}_1, \mathbf{M})}\, \mathbb{E}_{\mathbf{z}, \boldsymbol{\epsilon}}\, \mathbb{E}_{t \sim \mathcal{U}(0,1)} \left[\left\| \mathbf{u}_\theta(\mathbf{x}_t, t \mid \mathbf{M}) - \partial_t \mathbf{x}_t \right\|_2^2 \right]. \tag{11}$$

Training SynFlow is straightforward: we sample $(\mathbf{x}_1, \mathbf{M})$, a noise seed $\mathbf{z}$, and time t, then optimize the conditional velocity field with the pathwise objective in Eq. (11). This supervision, together with deterministic sampling, encourages long-range connectivity and yields interpretable trajectories in which topology typically stabilizes earlier than texture.

To effectively synthesize connectivity-consistent thin structures, we design a custom UNet-based architecture for SynFlow. Our architecture is driven by three targeted additions to a standard U-Net that make the conditioning highly effective:

Multi-Scale Mask Injection. We explicitly supply mask information at multiple matching spatial resolutions throughout the network (while time enters via a sinusoidal or MLP embedding). This ensures that structural geometry is injected and preserved at every scale of the feature hierarchy.

Lite-SPADE GroupNorm (LSGN). We propose the LSGN module to tightly couple normalization to the mask pyramid (Fig. 2, left). A small convolutional head maps the downsampled mask at each scale to spatial affine parameters that modulate group-normalized features:

$$\text{LSGN}(h \mid \mathbf{M}^{(s)}) \;=\; \gamma(\mathbf{M}^{(s)}) \odot \text{GN}(h) \;+\; \beta(\mathbf{M}^{(s)}). \tag{12}$$

This modulation keeps thin structures aligned through pooling and upsampling while remaining parameter-efficient.

Boundary-gating Module (BGM). We propose the BGM to prioritize the narrow band where topological errors originate (Fig. 2, right). A soft edge map

is derived from the mask using a morphological gradient and passed through a 3×3 convolution plus sigmoid to obtain a spatial gate; decoder and skip features are then multiplicatively modulated,

$$h_{\text{out}}(\mathbf{p}) \;=\; G_B(\mathbf{p})\, h_{\text{in}}(\mathbf{p}), \tag{13}$$

By dynamically allocating capacity to high-curvature interfaces and discouraging bleed-through, BGM focuses the network on the most difficult boundary regions. Together, multi-scale mask conditioning, LSGN, and BGM concentrate modeling power exactly where connectivity is fragile, transforming the baseline UNet into an architecture specialized for thin structures. Supplementary Sec. C.2 further analyzes mask-injection location, showing that encoder+decoder injection improves FID/KID over decoder-only conditioning, with dataset-specific qualitative comparisons in Supplementary Figs. S5–S9.

Controllable mask generator. To steer SynFlow toward regimes where annotation is most expensive, we pair it with a controllable mask generator. For in-domain data, we train a small conditional flow-matching model whose class label encodes crack sparsity bins (e.g., 0–0.5%, 0.5–1%); sampling a bin yields a mask that SynFlow renders into an image–mask pair. For cross-domain robustness, we further enrich masks via deterministic morphology: area-controlled propagation, branch-length extension with matched width, and thin/thick variants derived from skeleton radii. These operations expose SynFlow to sparsity, thickness, and branching patterns that are rare or absent in the original datasets. SegFlow and SynFlow are coupled at the data level rather than trained end-to-end: SynFlow supplies topology-consistent image–mask pairs for SegFlow training, while end-to-end co-evolution remains future work.

4 Experiments

4.1 Experimental settings

Datasets. We evaluate FMS^2 on five standard datasets—three crack sets (CRACK500, CrackTree260, CrackLS315 [44, 49, 50]) and two vessel sets (DRIVE retinal fundus, XCAD coronary angiography [21, 35]). The crack datasets span handheld to controlled laser acquisition and provide expert binary masks; foreground sparsity is severe, with mean crack coverage of roughly 0.6%–2.5%. DRIVE comprises color fundus photographs with ophthalmologist vessel maps. XCAD contains coronary X-ray angiograms with curated arterial trees, capturing highly branched vasculature and contrast-flow variability typical of interventional imaging. Detailed dataset statistics, preprocessing, and train/validation/test splits are provided in Supplementary Sec. A and Table S1.

Comparison methods. We compare to six recent baselines spanning complementary modeling philosophies for thin structures. Mask2Former [6] is a general-purpose segmentation transformer, and SemFlow [38] unifies segmentation and synthesis via latent-space flow matching. DConnNet [42], FlowSDF [2], Topograph [20], and SCSegamba [17] introduce architecture or topology priors

tailored to thin structures, covering transformer-style generalists, rectified/flow objectives, and topology-aware designs for ultra-sparse targets.

Evaluation metrics. We report overlap-based accuracy and topological fidelity. Overlap-based accuracy uses mean IoU and Dice (F1), with IoU $= \frac{|\mathcal{P}\cap\mathcal{G}|}{|\mathcal{P}\cup\mathcal{G}|}$ and Dice $= \frac{2|\mathcal{P}\cap\mathcal{G}|}{|\mathcal{P}|+|\mathcal{G}|}$, where $\mathcal{P}$ and $\mathcal{G}$ are predicted and ground-truth foreground sets. Topology is assessed via clDice [32] and the Betti matching error $\mu_k^{\text{err}}(\mathcal{P},\mathcal{G})$ for $k \in \{0,1\}$ [36], which resolves the limitations of the Betti number error by taking the spatial correspondence of topological features into account.

Implementation details. All experiments use PyTorch 2.8.0 on two NVIDIA A100 GPUs with Adam [11]. Inputs are cropped/resized to 256×256. For each crack dataset we curate 500 crack-centered patches and split them 80/10/10 into train/val/test; for DRIVE we use the standard 20/20 split, and for XCAD we use 100/26 train/test images. All competing baselines are trained and evaluated using their official code on our data splits. SegFlow is trained for 42k iterations with batch size 4 and base learning rate 10^{-5}, with 5k-step linear warmup followed by cosine annealing (decay= 0.1). SynFlow uses the same schedule with batch size 4, base learning rate 5×10^{-5}, and 150k iterations, and is sampled with Dormand–Prince (`dopri5`) using 10–20 function evaluations at inference.

4.2 Comparison of SegFlow with SOTA Methods

Overall results. Table 1 compares SegFlow with six recent baselines across five crack/vessel datasets. SegFlow delivers the best overall performance, improving average overlap-based accuracy and connectivity-aware quality over the strongest prior models. On average, it increases mIoU from 0.511 to 0.599 (+17.2%) and reduces μ^{err} from 82.145 to 51.524 (−37.3%). Note that these results are obtained with SegFlow alone, and do not use the SynFlow-generated images at all. Thus, the comparison isolates the segmentation formulation itself: the gains over diffusion- and flow-based baselines stem from image-space trajectory supervision rather than from additional synthetic training data.

Across the crack datasets, SegFlow consistently improves overlap-based accuracy. Its clearest topological gain appears on CrackTree260, where it reduces μ^{err} from 119.536 to 76.070, indicating fewer topology mismatches in complex branching cracks. On vessel datasets, SegFlow remains highly competitive and improves connectivity-aware quality; for example, on DRIVE it increases clDice from 0.780 to 0.830, reflecting more continuous vessel trees with fewer breaks. SegFlow also shows low run-to-run variation; full mIoU/μ^{err} mean ± standard deviation results are reported in Supplementary Table S2.

Latent-space unified generative models vs. image-space transport. Unified segmentation–generation frameworks such as SemFlow [38] learn a latent-space flow that must simultaneously preserve geometry and model appearance. On ultra-sparse, one-pixel targets, this coupling is brittle and often washes out thin structures, leading to low overlap and large topological mismatch (high μ^{err} in Table 1). In contrast, SegFlow performs explicit *image-space* transport from image to mask with trajectory-level supervision, so connectivity is enforced throughout

Table 1: Comparison of SegFlow with six SOTA methods across five datasets. Best results are in **bold**, and second-best results are underlined.

Method	Venue	CRACK500				CrackTree260				CrackLS315			
		Overlap ↑		Topological		Overlap ↑		Topological		Overlap ↑		Topological	
		mIoU	F1	clDice↑	μ^{err} ↓	mIoU	F1	clDice↑	μ^{err} ↓	mIoU	F1	clDice↑	μ^{err} ↓
Mask2Former	CVPR 2022	0.562	0.710	0.821	27.868	0.332	0.497	0.552	119.536	0.308	0.467	0.522	**93.310**
DConnNet	CVPR 2023	0.571	0.720	0.821	**23.651**	0.341	0.506	0.546	252.225	0.295	0.451	0.481	120.670
SemFlow	NeurIPS 2024	0.172	0.286	0.287	154.372	0.045	0.085	0.085	338.955	0.041	0.079	0.078	142.060
FlowSDF	IJCV 2025	0.531	0.687	0.763	36.558	0.577	0.728	0.729	157.830	0.221	0.358	0.358	137.150
Topograph	ICLR 2025	0.471	0.619	0.740	43.860	0.184	0.308	0.272	161.975	0.114	0.203	0.186	180.210
SCSegamba	CVPR 2025	0.549	0.702	0.794	24.791	0.437	0.604	0.609	202.800	0.386	0.548	0.557	94.130
SegFlow	**Ours**	**0.612**	**0.753**	**0.832**	24.884	**0.645**	**0.771**	**0.772**	**76.070**	**0.442**	**0.603**	**0.603**	96.260

Method	Venue	DRIVE				XCAD				Mean (5 datasets)			
		Overlap ↑		Topological		Overlap ↑		Topological		Overlap ↑		Topological	
		mIoU	F1	clDice↑	μ^{err} ↓	mIoU	F1	clDice↑	μ^{err} ↓	mIoU	F1	clDice↑	μ^{err} ↓
Mask2Former	CVPR 2022	0.444	0.615	0.661	280.480	0.616	0.761	0.812	15.923	0.452	0.610	0.674	107.423
DConnNet	CVPR 2023	0.567	0.722	0.780	**48.104**	0.636	0.776	**0.835**	**9.827**	0.482	0.635	0.693	90.895
SemFlow	NeurIPS 2024	0.295	0.454	0.449	149.208	0.157	0.269	0.225	148.942	0.142	0.235	0.225	186.707
FlowSDF	IJCV 2025	0.635	0.776	0.776	130.313	0.592	0.742	0.780	18.125	0.511	0.658	0.681	95.995
Topograph	ICLR 2025	0.553	0.711	0.676	239.062	0.536	0.696	0.676	120.500	0.372	0.507	0.510	149.121
SCSegamba	CVPR 2025	0.605	0.753	0.765	65.563	0.564	0.720	0.759	23.442	0.508	0.665	0.697	82.145
SegFlow	**Ours**	**0.653**	**0.789**	**0.830**	49.792	**0.642**	**0.780**	0.831	10.615	**0.599**	**0.739**	**0.774**	**51.524**

the ODE evolution rather than only at the final prediction. This difference is especially pronounced on topology-fragile crack data such as CrackTree260. Equally importantly, SegFlow decouples segmentation from synthesis: SegFlow handles deterministic image→mask transport, while SynFlow is used only as a mask→image renderer for label-efficient training. This separation avoids forcing a single latent trajectory to preserve appearance realism and sub-pixel topology.

Effect of training loss on topological fidelity. Architecture is one route to improving thin-structure segmentation (Table 1); Table 2 isolates an orthogonal route: supervision. All variants in Table 2 use the same U-Net backbone and training protocol; only the supervision changes. Topology-aware end-state losses reduce some failure modes, but they still supervise only the final prediction. SegFlow's flow-matching objective instead supervises the entire transport trajectory, yielding the best mean μ^{err} and the lowest μ^{err} on four of five datasets. Although clCE improves the mean μ^{err} from 82.145 (SCSegamba in Table 1) to 71.348, it still leaves a large gap to SegFlow's 51.524, suggesting that trajectory-level supervision is a stronger inductive bias for preserving thin-structure connectivity.

Table 2: μ^{err} comparison of supervision variants using the same U-Net backbone and training protocol as SegFlow. Only the supervision loss differs. Lower is better. Best results are in **bold**, and second-best results are underlined.

Method	Venue	CRACK500	CrackTree260	CrackLS315	DRIVE	XCAD	Mean
clCE	MICCAI	25.163	166.415	**92.700**	55.479	16.982	71.348
Skeleton Recall	ECCV 2024	26.779	144.850	92.930	85.898	18.231	73.738
SegFlow	**Ours**	**24.884**	**76.070**	96.260	**49.792**	**10.615**	**51.524**

Number of integration steps. Figure 3 studies the effect of SegFlow's Euler integration budget. Too few steps cause underfitting and under-developed masks, leading to reduced overlap and higher topological error. Increasing the steps improves both mIoU and μ^{err} until performance saturates. This demonstrates that SegFlow is robust to precise step tuning beyond the low-budget regime,

Table 3: Ablation of self-conditioning (SC) in SegFlow across five datasets.

Metric / Variant	CRACK500	CrackTree260	CrackLS315	DRIVE	XCAD
↑ mIoU w/o SC	0.612	0.645	0.442	0.653	0.642
↑ mIoU w/ SC	**0.628**	**0.654**	**0.453**	**0.664**	**0.650**
↓ μ^{err} w/o SC	24.88	76.07	96.26	49.79	10.62
↓ μ^{err} w/ SC	**21.26**	**73.04**	**92.91**	**41.68**	**9.68**

motivating our fixed, small-step budget to balance accuracy and compute. Additional transport-path and inference-cost analyses are provided in Supplementary Secs. B.2 and B.3.

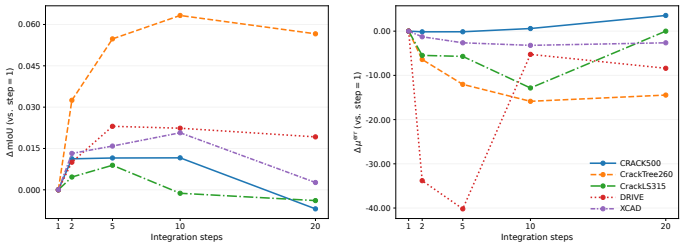

Fig. 3: Effect of integration steps on segmentation performance. We report mIoU and μ^{err} changes relative to step=1 across datasets.

Self-conditioning. Table 3 evaluates an optional self-conditioning input that feeds a running terminal-mask estimate back into the velocity network. Across all five datasets, this improves mIoU and reduces μ^{err}, indicating fewer topology-breaking artifacts during transport. These gains are most pronounced on sparse crack datasets where minor discontinuities quickly fragment centerlines. As a lightweight modification, self-conditioning provides a practical way to stabilize ultra-thin structures with minimal complexity.

Ablation of LSGN and BGM. Table 4 isolates the contribution of the two topology-aware injection components. Removing either component consistently weakens both overlap and topology metrics, while also increasing FID. The full configuration achieves the strongest results because LSGN supplies mask-conditioned spatial modulation across scales, whereas BGM concentrates capacity near topology-critical boundaries.

Table 4: Ablation of LSGN and BGM on CRACK500 and CrackTree260. Higher is better for mIoU, F1, and clDice; lower is better for μ^{err} and FID.

Components		CRACK500					CrackTree260				
LSGN	BGM	mIoU↑	F1↑	clDice↑	μ^{err} ↓	FID↓	mIoU↑	F1↑	clDice↑	μ^{err} ↓	FID↓
×	✓	0.593	0.737	0.818	30.2	26.44	0.567	0.710	0.710	97.2	26.68
✓	×	0.581	0.727	0.803	34.2	30.48	0.562	0.706	0.707	94.3	28.80
✓	✓	**0.613**	**0.754**	**0.835**	**28.0**	**18.36**	**0.603**	**0.739**	**0.740**	**88.0**	**20.14**

Qualitative comparison. We qualitatively analyze SegFlow in Figure 4 on representative crack and vessel examples. Across the crack rows, competing methods either miss thin side branches or introduce spurious fragments (yellow

and green arrows), whereas SegFlow recovers nearly all annotated branches with clean, one-pixel traces and intact junctions. On DRIVE and XCAD, transformer- and Mamba-based methods tend to break peripheral vessels or hallucinate wisps around dense trees, while SegFlow yields coherent arterial networks that follow vessel curvature more faithfully. We omit SemFlow and Topograph from Figure 4, as their predictions collapse on the thinnest crack datasets (Topograph on CRACK500/CrackLS315, SemFlow on all five datasets; see Table 1). For SegFlow, additional qualitative examples and representative failure cases are shown in Supplementary Figs. S1–S2.

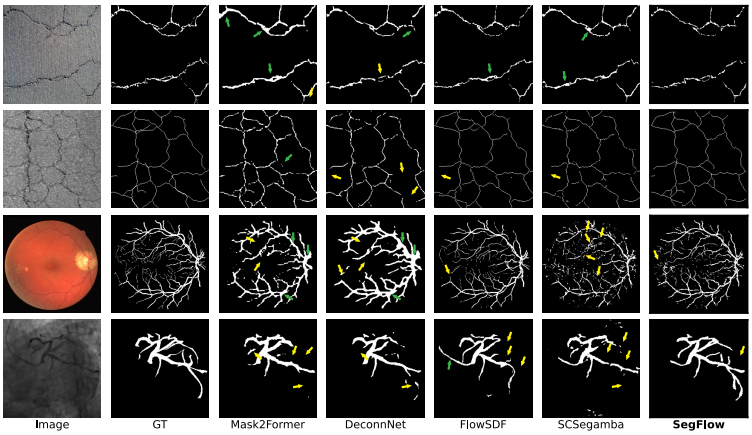

Fig. 4: Visual comparison of SegFlow against SOTA methods. Yellow arrows denote missed structures (FNs), green arrows indicate spurious predictions (FPs).

Label efficiency with partial real supervision. Before using SynFlow for augmentation, we evaluate the fidelity of generated masks and SynFlow-rendered, mask-conditioned images using FID (Table 5), computed for real vs. synthetic masks and real vs. SynFlow-rendered images. Figure 5 provides qualitative side-by-side examples, with additional controllable-mask and SynFlow rendering examples in Supplementary Figs. S3–S4.

Figure 6 evaluates how SynFlow-generated pairs can substitute for part of the expert-annotation budget. We train SegFlow with only $0.25R$ labeled images (where R is the full real training set) and progressively add SynFlow-generated, pixel-aligned image–mask pairs whose masks are sampled from the controllable mask generator to target diverse sparsity and topology regimes; implementation details and examples are provided in Supplementary Sec. C.1. With $0.25R$ alone, both IoU and clDice drop noticeably compared to full supervision R. As more synthetic pairs are added ($0.25R$+$4S \rightarrow 0.25R$+$16S \rightarrow 0.25R$+$64S$), performance improves consistently and, on the crack datasets, largely closes the gap to R. On XCAD, SynFlow also yields clear gains over $0.25R$, although a residual gap to full supervision remains, reflecting the higher appearance variability of coronary angiography. Overall, these trends indicate that SynFlow can markedly reduce the amount of expensive pixel-level annotation needed to

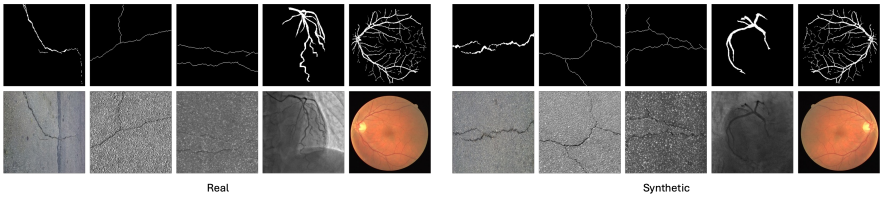

Fig. 5: Real vs. SynFlow-generated mask–image pairs. SynFlow preserves mask–image alignment while producing realistic textures; additional controllable-mask and rendering examples are shown in Supplementary Figs. S3–S4.

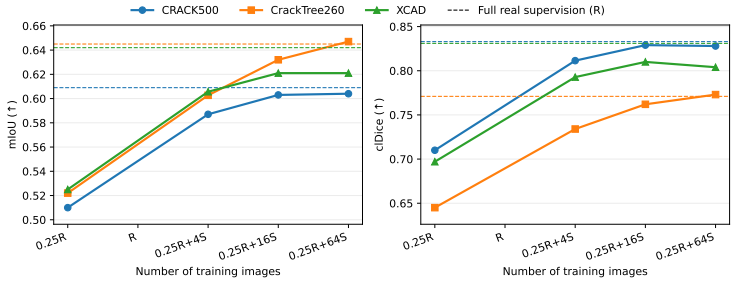

Fig. 6: Effect of SynFlow synthesis on label efficiency. SegFlow performance when training with a small portion of real labels ($0.25R$) and progressively adding SynFlow images ($0.25R$+$4S$, $0.25R$+$16S$, $0.25R$+$64S$). Here R is the full real training set (R=400 for Crack500 and CrackTree260, R=100 for XCAD) and S= $0.25R$; the dashed line marks full real supervision at R. Left: IoU, right: clDice.

reach strong performance, making it a practical data-centric tool for label-efficient thin-structure segmentation.

Table 5: FID between real and SynFlow-generated masks/images across five datasets. Lower values indicate better distributional fidelity.

Metric	CRACK500	CrackTree260	CrackLS315	DRIVE	XCAD
Mask FID	25.67	15.68	13.84	15.04	76.02
Image FID	18.36	20.14	19.81	3.04	40.13

Cross-domain robustness under domain shift. Table 7 evaluates source-only transfer across the three crack datasets, where changes in background texture, noise, and acquisition conditions often fragment or erase thin structures and lead to substantial generalization drops. Although SegFlow already outperforms the strongest architecture baselines in-domain (Table 1), performance still degrades under domain shift. Classic DA (i.e., horizontal/vertical flips, rotations, cropping, brightness–contrast jitter, Gaussian noise, and Gaussian blur) provides only modest benefits, improving mIoU by about +0.006 (+2.4% relative gain) on average across the three cross-domain experiments. SynFlow yields larger improvements, increasing mIoU by +0.109 on average across the same three experiments, corresponding to about +46.6% relative gain. This gap arises because SynFlow differs fundamentally from standard DA. Rather than enforcing invariance via label-preserving transforms, SynFlow is a paired mask→image generator that

Table 6: Structural distribution gap between CRACK500 (C) and CrackTree260 (T), measured by bidirectional KL divergence. Lower values indicate structural alignment.

Structural Measure	Original Training Datasets		w/ SynFlow Samples	
	C→T	T→C	C→T	T→C
Crack density	3.965	1.265	**1.086**	**0.716**
Crack length	0.033	0.187	**0.012**	**0.013**
Branch points	4.337	2.716	**0.141**	**0.189**

Table 7: Cross-domain generalization across three crack datasets. Each cell reports mIoU/F1. Values in parentheses indicate relative gains over SegFlow.

Method	CRACK500→ CrackTree260	CrackLS315→ CRACK500	CrackTree260→ CRACK500
SCSegamba	0.224/0.363	0.212/0.339	0.246/0.384
FlowSDF	0.212/0.347	0.195/0.320	0.247/0.386
SegFlow	0.251/0.396	0.223/0.352	0.254/0.393
SegFlow+DA	0.259/0.408 (+3.2%/+3.0%)	0.229/0.358 (+2.7%/+1.7%)	0.257/0.395 (+1.2%/+0.5%)
SegFlow+SynFlow	**0.305/0.444** (+21.5%/+12.1%)	**0.411/0.566** (+84.3%/+60.8%)	**0.340/0.489** (+33.9%/+24.4%)

produces *pixel-aligned* image–mask samples. Using controllable mask synthesis that varies sparsity, width, and connectivity, together with the renderer's ability to model realistic texture and appearance variation conditioned on these masks, it introduces structured, topology-relevant shifts in the source domain that better match cross-domain failure modes. This suggests that SynFlow improves transfer primarily by expanding the source distribution structurally/topologically, not merely stylistically: the generated pairs vary crack density, length, width, and branching while preserving pixel-level alignment. As shown in Table 6, adding SynFlow samples reduces the bidirectional structural distribution gap between CRACK500 and CrackTree260, supporting the cross-domain gains in Table 7.

5 Conclusion

We present FMS^2, a dual-module flow-matching framework for thin-structure learning. SegFlow performs explicit pixel-space image-to-mask transport with trajectory-level supervision, improving both overlap and topology across five crack and vessel datasets; in mean performance, mIoU increases from 0.511 to 0.599 and μ^{err} decreases from 82.145 to 51.524. SynFlow complements this segmentation pathway with topology-consistent mask-to-image rendering, providing pixel-aligned synthetic supervision that recovers near-full performance from 25% real annotations and improves cross-domain robustness by expanding structural/topological diversity. We release 10k crack and 1k vessel image–mask pairs to support future benchmarking and generalizable thin-structure models.

Acknowledgments. The first author gratefully acknowledges support from the University of Illinois Urbana-Champaign Graduate College through the 2025–2026 Dissertation Completion Fellowship award, and thanks Prof. Svetlana Lazebnik for helpful suggestions and feedback that improved the presentation of this work.

References

1. Acebes, C., Moustafa, A.H., Camara, O., Galdran, A.: The Centerline-Cross Entropy Loss for Vessel-Like Structure Segmentation: Better Topology Consistency Without Sacrificing Accuracy. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI) 2024. pp. 710–720 (2024). https://doi.org/10.1007/978-3-031-72111-3_67
2. Bogensperger, L., Narnhofer, D., Falk, A., Schindler, K., Pock, T.: Flowsdf: Flow matching for medical image segmentation using distance transforms. International Journal of Computer Vision **133**(7), 4864–4876 (2025)
3. Chen, C., Chuah, J.H., Ali, R., Wang, Y.: Retinal Vessel Segmentation Using Deep Learning: A Review. IEEE Access **9**, 111985–112004 (2021). https://doi.org/10.1109/ACCESS.2021.3102176
4. Chen, J., Mei, J., **(others)**, Zhou, Y.: TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers. Medical Image Analysis **97**, 103280 (2024). https://doi.org/10.1016/j.media.2023.103280
5. Chen, R.T.Q., Lipman, Y., Ben-Hamu, H.: Flow matching for generative modeling (neurips 2024 tutorial). https://aisecure.github.io/flow-matching-tutorial/ (2024), tutorial website; accessed 2025-11-13
6. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A.: Masked-Attention Mask Transformer for Universal Image Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1290–1299 (2022). https://doi.org/10.1109/CVPR52688.2022.00136
7. Dong, S., Zhu, X., Chen, H., Zheng, Y., Frangi, A.F.: coDice: Connectivity-Preserving Dice Loss for 2D/3D Tubular Structure Segmentation. In: SPIE Medical Imaging. vol. 12529, p. 1252918 (2025). https://doi.org/10.1117/12.2651373
8. Gupta, S., Zhang, Y., Hu, X., Prasanna, P., Chen, C.: Topology-Aware Uncertainty for Image Segmentation. Advances in Neural Information Processing Systems (NeurIPS) **36** (2023)
9. Hu, X., Li, F., Samaras, D., Chen, C.: Topology-preserving deep image segmentation. Advances in neural information processing systems **32** (2019)
10. Isensee, F., Jäger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**(2), 203–211 (2021)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
12. Kirchhoff, Y., Heinrich, M.P.: Skeleton Recall Loss for Connectivity Conserving and Resource Efficient Segmentation of Thin Tubular Structures. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 218–234 (2024). https://doi.org/10.1007/978-3-031-41606-4_13
13. Lei, Q., Zhong, J., Dai, Q.: Enriching information and preserving semantic consistency in expanding curvilinear object segmentation datasets. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 233–250. Springer (2024)
14. Lei, Q., Zhong, J., Dai, Q., Hu, Y., Bai, Y., Li, X., Yao, Y.: Enriching information and preserving semantic consistency in expanding curvilinear object segmentation datasets. In: European Conference on Computer Vision (ECCV) (2024)
15. Lesage, D., Angelini, E.D., Bloch, I., Funka-Lea, G.: A review of 3D vessel lumen segmentation techniques: models, features and extraction schemes. Medical Image Analysis **13**(6), 819–845 (2009). https://doi.org/10.1016/j.media.2009.07.011

16. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
17. Liu, H., Jia, C., Shi, F., Xu, C., Chen, S.: Scsegamba: Lightweight structure-aware vision mamba for crack segmentation in structures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
18. Liu, J., Li, D., Xu, X., **(others)**: MambaHRNet: Enhanced High-Resolution Network for Bridge Damage Segmentation. *submitted* (2024)
19. Liu, Y., Gong, C., Kong, L., Wang, Q., Lu, J.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: International Conference on Learning Representations (ICLR) (2023)
20. Lux, L., Berger, A.H., Weers, A., Stucki, N., Rückert, D., Bauer, U., Paetzold, J.C.: Topograph: An efficient graph-based framework for strictly topology preserving image segmentation. In: International Conference on Learning Representations (ICLR) (2025)
21. Ma, Y., Hua, Y., Deng, H., Song, T., Wang, H., Xue, Z., Cao, H., Ma, R., Guan, H.: Self-supervised vessel segmentation via adversarial learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7536–7545 (2021)
22. Mosinska, A., Marquez-Neila, P., Koziński, M., Fua, P.: Beyond the pixel-wise loss for topology-aware delineation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3136–3145 (2018)
23. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4296–4304 (2024)
24. Mou, L., Zhao, Y., Fu, H., Liu, Y., Cheng, J., Zheng, Y., Su, P., Yang, J., Chen, L., Frangi, A.F., Akiba, M., Liu, J.: CS2-Net: Deep learning segmentation of curvilinear structures in medical imaging. Medical Image Analysis **67**, 101874 (2021). https://doi.org/10.1016/j.media.2020.101874
25. Ouyang, Y., Kuang, X., Xiong, M., Wang, Z., Wang, Y.: A Novel Hybrid Approach for Retinal Vessel Segmentation with Dynamic Long-Range Dependency and Multi-Scale Retinal Edge Fusion Enhancement. Pattern Recognition (2025). https://doi.org/10.1016/j.patcog.2024.110387
26. Qi, L., Yang, L., Guo, W., Xu, Y., Du, B., Jampani, V., Yang, M.H.: UniGS: Unified Representation for Image Generation and Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6305–6315 (2024). https://doi.org/10.1109/CVPR.2024.00638
27. Rajitha, K.V., Prasad, K., Yegneswaran, P.P.: Segmentation and Classification Approaches of Clinically Relevant Curvilinear Structures: A Review. Journal of Medical Systems **47**(1), 40 (2023). https://doi.org/10.1007/s10916-023-01927-2
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (2022). https://doi.org/10.1109/CVPR52688.2022.01046
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer (2015)

30. Schusterbauer, J., Gui, M., Fundel, F., Ommer, B.: Diff2flow: Training flow matching models via diffusion model alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
31. Shi, Y., **(others)**: Deep Learning in Crack Detection: A Comprehensive Review. *** (in press, 2025)*** (2025)
32. Shit, S., Paetzold, J.C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J.P.W., Bauer, U., Menze, B.H.: cldice: A novel topology-preserving loss function for tubular structure segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16560–16569 (2021). https://doi.org/10.1109/CVPR46437.2021.01629
33. Song, Y., Kang, J., Su, Y., Zhang, S., Zhang, Q., Yu, Y., Zhan, Z., Zhang, W.: MambaFuse: Cross-scale state space fusion for crack segmentation. Developments in the Built Environment **24**, 100751 (2025). https://doi.org/10.1016/j.dibe.2023.100751
34. Song, Y., Su, Y., Zhang, S., Wang, R., Yu, Y., Zhang, W., Zhang, Q.: CrackdiffNet: A Novel Diffusion Model for Crack Segmentation and Scale-Based Analysis. Buildings **15**(11), 1872 (2025). https://doi.org/10.3390/buildings15111872
35. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging **23**(4), 501–509 (2004). https://doi.org/10.1109/TMI.2004.825627
36. Stucki, N., Paetzold, J.C., Shit, S., Menze, B., Bauer, U.: Topologically faithful image segmentation via induced matching of persistence barcodes. In: International Conference on Machine Learning. pp. 32698–32727. PMLR (2023)
37. Van Gansbeke, S., De Brabandere, B.: A simple latent diffusion approach for panoptic segmentation and mask inpainting. In: Computer Vision – ECCV 2024. pp. 271–290. Lecture Notes in Computer Science, Springer (2024)
38. Wang, C., Li, X., Qi, L., Ding, H., Tong, Y., Yang, M.H.: SemFlow: Binding Semantic Segmentation and Image Synthesis via Rectified Flow. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
39. Wu, C.H., Chen, S.H., Hu, C.Y., Wu, H.Y., Chen, K.H., Chen, Y.Y., Su, C.H., Lee, C.K., Liu, Y.L.: DeNVeR: Deformable Neural Vessel Representations for Unsupervised Video Vessel Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
40. Wu, H., Li, W., Ye, Y., Tang, Y., Chen, J., **(others)**: MedSegDiff-V2: Diffusion-Based Medical Image Segmentation with Transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024)
41. Xu, Q., Ma, Z., He, N., Duan, W.: DCSAU-Net: A deeper and more compact split-attention U-Net for medical image segmentation. Computers in Biology and Medicine **154**, 106626 (2023). https://doi.org/10.1016/j.compbiomed.2023.106626
42. Yang, Z., Farsiu, S.: Directional Connectivity-based Segmentation of Medical Images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11525–11535 (2023). https://doi.org/10.1109/CVPR.2023.01167
43. Zavadski, R., Feiden, W., Rother, C.: Controlnet-xs: Spotting shortcut learning in controlled image generation. In: European Conference on Computer Vision (ECCV). pp. 4511–4522 (2024)
44. Zhang, L., Yang, F., Zhang, Y.D., Zhu, Y.J.: Road crack detection using deep convolutional neural network. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 3708–3712 (2016)

45. Zhang, L., Rao, A., Agrawala, M.: Adding Conditional Control to Text-to-Image Diffusion Models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3836–3847 (2023). https://doi.org/10.1109/ICCV.2023.03885
46. Zhang, Y., Yu, P., Zhu, Y., Chang, Y., Gao, F., Wu, Y.N., Leong, O.: Flow priors for linear inverse problems via iterative corrupted trajectory matching. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
47. Zhou, S., Canchila, C., Song, W.: Deep learning-based crack segmentation for civil infrastructure: data types, architectures, and benchmarked performance. Automation in Construction **146**, 104623 (2023). https://doi.org/10.1016/j.autcon.2023.104623
48. Zhou, Y., Li, L., Che, Z.: nnWNet: Rethinking the Use of Transformers in Biomedical Image Segmentation and Calling for a Unified Evaluation Benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20852–20862 (2025)
49. Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S.: Cracktree: Automatic crack detection from pavement images. Pattern Recognition Letters **33**(3), 227–238 (2012). https://doi.org/10.1016/j.patrec.2011.11.004
50. Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., Wang, S.: Deepcrack: Learning hierarchical convolutional features for crack detection. IEEE Transactions on Image Processing **28**(3), 1498–1512 (2019). https://doi.org/10.1109/TIP.2018.2878966
51. Zuo, X., Sheng, Y., Shen, J., Shan, Y.: Topology-aware Mamba for Crack Segmentation in Structures. Automation in Construction **168, Part A**, 105845 (2024). https://doi.org/10.1016/j.autcon.2023.105845