CRACKINSTSYNTH: TOPOLOGY-AWARE GENERATIVE DATA-AUGMENTATION FRAMEWORK FOR CRACK INSTANCE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Instance-level crack segmentation is critical for automated structural health monitoring of tunnels and bridges, yet progress is limited by the scarcity of densely annotated datasets with instance-level labels. To address this gap, we make two key contributions. First, we introduce CrackInst1K, to our knowledge the first publicly available instance-level crack segmentation dataset, comprising 1025 high-resolution tunnel images with pixel-accurate instance masks. Second, we propose CrackInstSynth, a generative data-augmentation framework that substantially enlarges instance-level crack corpora while preserving geometric and topological realism. CrackInstSynth integrates three coordinated modules: (i) Region-level Instance Placement (RIP), which partitions the canvas into quadrants to strategically position crack instances for diverse layouts; (ii) a Physics-driven Skeleton Generator (PSG) that enriches morphological variability by growing crack skeletons via physical simulation; and (iii) a Topology-Preserving Generation Module (TPGM) that employs a two-stage conditional diffusion pipeline (skeleton→mask, mask→image) to produce paired, width-aware instance masks and corresponding images while enforcing intra-instance topology and inter-instance separation. Extensive experiments show that augmenting real data with CrackInstSynth consistently improves the performance of multiple instance segmentation models on CrackInst1K and other benchmarks, validating both visual fidelity and downstream effectiveness. We will release CrackInst1K and CrackInstSynth to facilitate future research in structural health monitoring.

1 INTRODUCTION

Cracks on the surfaces of tunnels Lei et al. (2024c); Wang et al. (2023), bridges Lei et al. (2024c); Chen et al. (2022), pavements Lei et al. (2023; 2024a); Chen et al. (2025), and other civil infrastructure are critical indicators of material fatigue, water ingress, and progressive structural deterioration Yuan et al. (2024). Early and accurate detection enables preventive maintenance before minor defects escalate into serious hazards, thereby extending service life and reducing life-cycle costs Panella et al. (2022). For tunnel linings in particular, unchecked crack growth can quickly compromise structural integrity, whereas timely repair markedly reduces subsequent expenditures Wang et al. (2023).

While binary crack maps suffice for coarse assessment, downstream tasks—such as measuring per-crack size, prioritizing repair schedules, and tracking temporal evolution—require distinguishing individual cracks Zhao et al. (2024a); Lei et al. (2024c). Instance-level segmentation simultaneously localizes and uniquely labels each crack, yielding pixel-accurate masks that support geometric analysis, path tracing, and longitudinal studies. Compared with semantic segmentation, which merges all crack pixels into a single class, or detection pipelines that output only bounding boxes, instance segmentation provides the level of detail needed for automated, priority-driven maintenance strategies Shi et al. (2021). Fig. 1 illustrates a workflow for measuring tunnel-lining crack properties using instance segmentation.

Despite recent progress, crack segmentation research remains severely data-constrained. Public benchmarks such as CFD Shi et al. (2016), DeepCrack Zou et al. (2018), and CRACK500 Shi

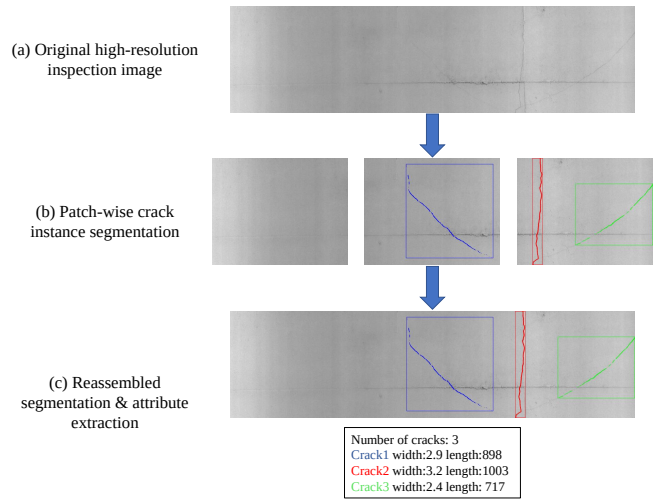

Figure 1: Workflow for instance-level crack segmentation and measurement. (a) Original high-resolution inspection image. (b) Patch-wise instance segmentation: the image is tiled into patches, and each patch is processed by an instance segmentation model to produce individual crack masks (blue, red, and green). (c) Reassembled segmentation and attribute extraction: patch-wise predictions are mapped back to the original image coordinates and merged to obtain image-level crack instances, from which geometric attributes (e.g., width and length) are computed and summarized.

et al. (2016) contain only a few hundred images, and collecting additional data is costly: a single tunnel survey typically yields fewer than 500 usable frames, while pixel-level annotation by safety experts can take several minutes per image Panella et al. (2022); Liu et al. (2019); Rezaie et al. (2020). The shortage is even more acute for instance-level labels: no open dataset currently provides pixel-accurate instance IDs (unique labels for each crack), and most studies rely on binary masks or sparsely annotated subsets Zhao et al. (2024a). This gap hampers automatic structural health monitoring and highlights the need for cost-effective methods to enlarge corpora for crack instance segmentation.

To mitigate the twin shortages of scale and instance granularity, we curate **CrackInst1K**, a publicly available tunnel-lining crack dataset comprising 1025 high-resolution images (1024×1024 pixels). Images collected across multiple years and tunnels are first tiled into patches and then randomly mosaicked to ensure de-identification while guaranteeing that each image contains at least one annotated crack instance. At a scale where $\sim$1,000 pixels correspond to $\sim$1,m on the lining, each image covers roughly 1,m^2. Compared with prior crack segmentation benchmarks, CrackInst1K provides explicit instance labels together with accurate geometric scale and focuses on two crack types that require precise sizing in tunnel and bridge structural health monitoring—hairline cracks and branched/intersecting cracks—while excluding map cracking (crazing), generally does not require exact dimensional measurement. This establishes a fine-grained reference set for geometry-accurate, instance-level evaluation.

Building on this resource, we develop **CrackInstSynth**, a topology-aware generative augmentation framework that produces diverse crack instance segmentation image–mask pairs without additional manual labeling. The pipeline first performs **Region-level Instance Placement (RIP)** by selecting one to three seed instances from CrackInst1K and placing their masks at random positions within a randomly selected canvas quadrant. Next, a **Physics-driven Skeleton Generator (PSG)** stochastically grows each seed's skeleton under a physics-based crack growth simulation, thereby increasing the informationtiveness of seeds. The grown seeds are then processed by a **Topology-Preserving Generation Module (TPGM)** with two diffusion stages: (i) a skeleton$\rightarrow$mask stage, in which a pixel-space generative model conditionally produces width-aware instance masks; and (ii) a mask$\rightarrow$image stage, in which a topology-consistent, ControlNet-style (TC-ControlNet) diffusion model, conditioned on the width-aware masks and a predefined text prompt, renders photorealistic crack images whose topology exactly matches the conditioning masks. By integrating these three

modules, CrackInstSynth generates diverse, label-aligned samples that augment real data and enable large-scale training of crack instance segmentation models.

We assess CrackInstSynth on CrackInst1K as well as the public DeepCrack Liu et al. (2019) and CRACK500 Shi et al. (2016) benchmarks. The evaluation covers both the visual fidelity of the synthetic images and the performance gains of the synthetic image-mask pairs provide when training state-of-the-art instance segmentation models. In every case, the augmented data yields clear improvements over baseline training sets, confirming the practical value of the proposed framework for crack analysis. We will release CrackInst1K and CrackInstSynth to support future research.

In summary, this paper makes the following contributions:

1. **CrackInst1K**: a publicly available dataset of 1025 tunnel-lining images (1024×1024) with pixel-accurate per-instance masks. Images are de-identified via tile–mosaic preprocessing; the set focuses on hairline and branched/intersecting cracks, forming a fine-grained benchmark.
2. **CrackInstSynth**: a topology-aware generative augmentation framework that produces diverse, label-aligned image–mask pairs without extra labeling, integrating RIP, PSG, and TPGM with a two-stage diffusion pipeline (skeleton→mask, mask→image).

2 Related Work

2.1 Crack Instance Segmentation

Most deep learning studies on cracks still treat the problem as binary or semantic segmentation, classifying all crack pixels as a single class. Transformer-based detectors such as CrackFormer Liu et al. (2021) improve hairline preservation by modeling long-range context, yet intersecting cracks often remain merged, which limits geometric analysis that requires per-crack masks.

More recent work adapts generic instance frameworks to the thin-object regime of cracks. Mask R-CNN He et al. (2017) pipelines augmented with morphological closing reconnect fragmented masks and improve accuracy on tunnel linings Huang et al. (2022). Orientation-aware detectors represent a curved crack as a sequence of rotated segments to separate crossings and branches Chen et al. (2023). Lightweight one-stage networks enhanced with multi-head and triplet attention achieve real-time instance segmentation while boosting recall for fine structures Yu et al. (2025). Other approaches advance per-crack labeling but still struggle with continuity and ambiguous boundaries Zhao et al. (2024b); Lei et al. (2024c).

Topology-preserving techniques address these weaknesses. A differentiable connectivity loss penalizes broken masks Pantoja-Rosero et al. (2022), and ambiguity-aware representation learning refines uncertain crack edges Chen et al. (2024). Large benchmarks such as OmniCrack30k Benz & Rodehorst (2024) expand training data, yet none provide dense instance labels with calibrated metric scale. CrackInst1K and the CrackInstSynth framework fill this gap by supplying precisely scaled instance annotations and generating additional topology-consistent data, enabling large-scale instance segmentation without additional manual labeling.

2.2 Generative Data Augmentation

Conditional GANs were the first practical engines for paired data synthesis. pix2pixHD and SPADE translate coarse semantic masks into photorealistic surfaces and have been adapted to crack and road-defect imagery Wang et al. (2018); Park et al. (2019). Such GAN-based pipelines boost realism but offer limited geometric diversity and often break thin structures, restricting their value for topology-sensitive tasks.

Recent augmentation studies explore alternative generative cues. Cut-and-paste strategies like InstaBoost warp foreground masks to create new layouts Fang et al. (2019), whereas domain-randomized renderers synthesize cracks by overlaying procedural textures on material maps Yang et al. (2020). MosaicFusion shows that a single diffusion pass can populate disjoint canvas regions with multiple labeled objects for detection and segmentation Xie et al. (2025), and Panoptic Diffusion embeds instance IDs in the latent space to reduce label misalignment Bansal et al. (2023). Although these

methods enlarge datasets efficiently, none explicitly maintain the connectivity and mutual separation required by elongated, intersecting cracks.

Latent diffusion models inject text or mask guidance into the denoising process, delivering higher fidelity and broader mode coverage Rombach et al. (2022). ControlNet Zhang et al. (2023), T2I-Adapter Mou et al. (2024), and related variants refine mask conditioning but still downsample masks and weaken structural cues. CrackInstSynth advances this line by coupling physics-driven skeleton growth with a two-stage, topology-preserving diffusion module, generating large volumes of geometry-faithful image–mask pairs tailored to crack instance segmentation.

3 METHODOLOGY

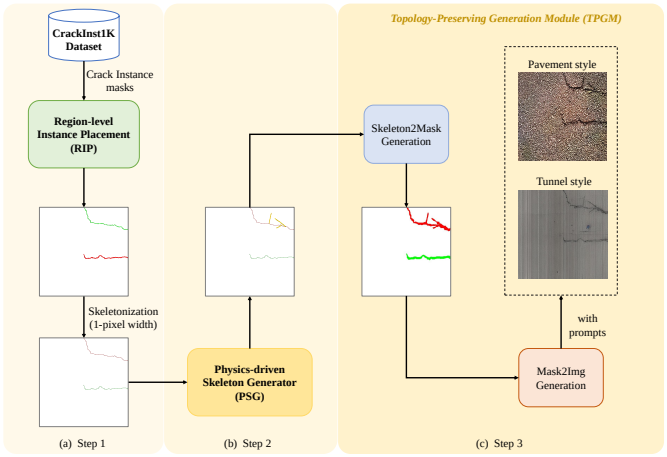

Figure 2: Overall workflow of the proposed CrackInstSynth framework. The pipeline comprises three sequential stages: (a) Region-level Instance Placement (RIP), (b) Physics-driven Skeleton Generator (PSG), and (c) Topology-Preserving Generation Module (TPGM), which performs Skeleton-to-Mask and Mask-to-Image synthesis while enforcing crack topology.

CrackInstSynth tackles the scarcity of crack-instance data by generating topology-consistent image–mask pairs in three stages, as shown in Fig. 2. **(a) Region-level Instance Placement (RIP)** selects one to three seed instances (masks) from the curated *CrackInst1K* dataset and placing their masks at random positions within a randomly selected canvas quadrant, producing diverse multi-instance layouts; the placed masks are then skeletonized to one-pixel width to prepare for physics-based growth. **(b) Physics-driven Skeleton Generator (PSG)** takes the one-pixel skeletons and stochastically grows each under a physics-based crack growth simulation, injecting physically plausible branching and increasing the informationtiveness. **(c) Topology-Preserving Generation Module (TPGM)** then runs a two-stage diffusion process: a *Skeleton2Mask* network inflates each skeleton into a width-aware instance mask, and a *Mask2Img* network—conditioning a topology-consistent, ControlNet-style diffusion model (TC-ControlNet) on the width-aware masks and style prompts (e.g., *pavement*, *tunnel*)—renders photorealistic crack images whose geometry and topology exactly match the conditioning masks.

3.1 REGION–LEVEL INSTANCE PLACEMENT

Let $\mathcal{D} = \{(\mathbf{M}_j, \mathbf{b}_j)\}$ denote the set of pixel masks $\mathbf{M}_j \subset [0,1]^{H_0 \times W_0}$ in *CrackInst1K* and their axis–aligned bounding boxes $\mathbf{b}_j = [x_j, y_j, w_j, h_j]$. Given a blank canvas $\mathcal{C} \in \mathbb{R}^{H \times W \times 3}$ of size $H = W = 1024$, RIP synthesises a layout with $n \sim \mathcal{U}\{1,2,3\}$ instances.

The canvas is partitioned into four non–overlapping quadrants $\mathcal{R} = \{R^1, R^2, R^3, R^4\}$, each of size 512×512. RIP selects n distinct regions $\{R^{\pi(1)}, \ldots, R^{\pi(n)}\}$ by a random permutation π, then places the n sampled masks after an i.i.d. translation

$$\begin{aligned} \mathbf{t}_i &\sim \mathcal{U}\Big([0,\, w_i^{\max}] \times [0,\, h_i^{\max}]\Big), \\ w_i^{\max} &= w_{R^{\pi(i)}} - w_j, \\ h_i^{\max} &= h_{R^{\pi(i)}} - h_j. \end{aligned} \tag{1}$$

so that every translated box $\tilde{\mathbf{b}}_i = \mathbf{b}_j + \mathbf{t}_i$ is fully contained in its region. The procedure returns a colour canvas $\mathbf{I}_{\text{RIP}}$, that is, a segmentation of multiple crack instances and the translated annotations $\{\tilde{\mathbf{M}}_i, \tilde{\mathbf{b}}_i\}$.

RIP can be summarized as follows: (i) we evenly divide a 1024×1024 canvas into four non-overlapping 512×512 regions, inspired by MosaicFusion Xie et al. (2025), to increase the information density per image; (ii) we then iteratively place up to three sampled crack instances at random into the four regions. This design encodes a civil-engineering prior: since *CrackInst1K* is calibrated such that $\sim$1000,px $\approx$ 1,m, a 1024×1024 canvas (about $1,\mathrm{m}^2$) should typically contain no more than three cracks.

Unlike MosaicFusion, in RIP the instances are sampled at the 1024×1024 scale and then placed into 512×512 regions. We allow region overflow at placement time: portions extending beyond the assigned region are preserved, whereas any content outside the outer 1024×1024 canvas is clipped. This relaxation increases layout diversity and retains potential cross-region interactions (e.g., intersecting cracks), while keeping the global canvas consistent.

3.2 Physics-driven Skeleton Generator

For each seed mask $\tilde{\mathbf{M}}_i$ produced by RIP we extract a one-pixel-wide skeleton $\mathbf{S}_i = \textsc{Thin}(\tilde{\mathbf{M}}_i)$ using Zhang–Suen thinningLam et al. (1992). Let $\mathcal{B}_i$ denote the axis-aligned bounding box of $\tilde{\mathbf{M}}_i$. PSG expands $\mathcal{B}_i$ by a scale factor $\alpha \in [1.2, 1.6]$ to obtain a *growth window* $\hat{\mathcal{B}}_i$ inside which a stochastic crack propagation process is simulated. Starting from k randomly sampled pixels on $\mathbf{S}_i$ ($k \sim \mathcal{U}\{1, \ldots, M_k\}$), default $M_k = 4$, a random walk Lei et al. (2024a) adds new skeleton points until a maximum relative length $m = 0.8$ of the window is reached or a step limit is met:

$$\mathbf{S}_i^{\text{new}} \;=\; \textsc{RandomWalk}\big(\mathbf{S}_i,\, \hat{\mathcal{B}}_i,\, k,\, m,\, \textit{min}, \textit{max}, \ell, \theta\big),$$

where *min* and *max* are the minimum/maximum step counts, ℓ is the step length in pixels (default 2) and θ bounds the turning angle ($\pm30^\circ$). The walk is confined to $\hat{\mathcal{B}}_i$ to avoid inter-instance overlap.

See Apendix A.2.1 for the Algorithm 1 of RandomWalk.

Physically, cracks in the infrastructure structures propagate along principal stress directions while exhibiting stochastic branching. Embedding this behaviour via bounded random walks injects *plausible curvature, length variation and side branches*, thereby expanding the geometric distribution beyond the limited shapes in CrackInst1K.

3.3 Topology-Preserving Generation Module (TPGM)

TPGM takes as input the PSG-augmented skeletons. Let $\{\mathbf{S}_i^{\text{new}}\}_{i=1}^{n}$ be the grown skeletons from PSG and define the *instance-ID skeleton map* $\mathbf{S}_{\text{PSG}} \in \{0, \ldots, n\}^{H\times W}$ by assigning value i to pixels on $\mathbf{S}_i^{\text{new}}$ and 0 to background. Each pixel of $\mathbf{S}_{\text{PSG}}$ encodes an *instance identifier* (0 for background, $1{:}n$ for cracks) along one-pixel-wide centerlines. TPGM produces a topology-aligned, *width-aware instance map* $\mathbf{M}_{\text{WA}} \in \{0, \ldots, n\}^{H\times W}$ that expands each skeleton to its physical width, and a photorealistic image $\mathbf{I}$, through two diffusion stages.

Formally, TPGM learns

$$\mathcal{F}:\; \mathbf{S}_{\text{PSG}} \;\longmapsto\; (\mathbf{M}_{\text{WA}},\, \mathbf{I}). \tag{2}$$

The mapping is realized by (i) a *Skeleton→Mask* diffusion network that inflates $\mathbf{S}_{\text{PSG}}$ into $\mathbf{M}_{\text{WA}}$; and (ii) a *Mask→Image* diffusion network that renders $\mathbf{I}$ conditioned on $\mathbf{M}_{\text{WA}}$ while honoring the above topological requirements.

3.3.1 STAGE 1: SKELETON→MASK DIFFUSION

Given the one-pixel-wide instance-ID map $\mathbf{S}_{\text{PSG}} \in \{0, \ldots, n\}^{H \times W}$, the goal is to infer a width-aware instance map $\mathbf{M}_{\text{WA}} \in \{0, \ldots, n\}^{H \times W}$ that satisfies connectivity and separation. We adopt the *pixel-level Semantic Diffusion Model (SDM)* framework Wang et al. (2022); Lei et al. (2024a), conditioning directly on $\mathbf{S}_{\text{PSG}}$. Skeleton conditioning is injected via SPADE Park et al. (2019) layers placed in every upsampling block of the UNet; no additional modalities are required.

Pixel-space diffusion preserves high-frequency details and has been shown to outperform latent-space diffusion (e.g., LDM) on tasks requiring strict structural fidelity, such as medical shape synthesis Konz et al. (2024) and curvilinear object augmentation Lei et al. (2024b). Because Skeleton→Mask uses only a single conditioning map without text prompts, the pixel-level SDM is more efficient and easier to train than latent counterparts while retaining full spatial resolution.

3.3.2 STAGE 2: MASK→IMAGE DIFFUSION

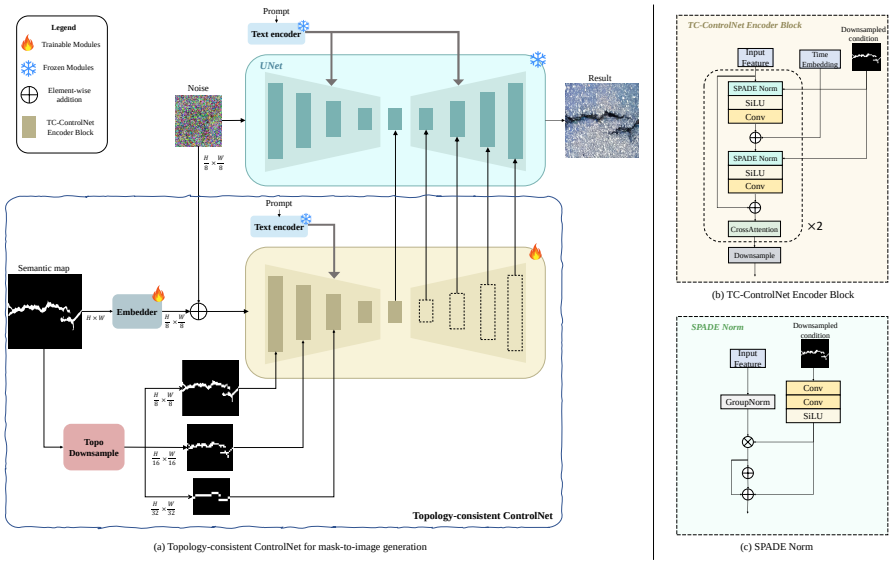

Figure 3: Topology-Consistent ControlNet (TC-ControlNet) for mask-to-image generation. (a) The upper path (blue) is the frozen Stable Diffusion UNet; the lower path (yellow) is the trainable ControlNet branch that injects multi-scale, topology-preserving mask features. A *TopoDownsample* module provides three scale masks whose region-adjacency graph is unchanged. (b) Encoder block details: the downsampled mask modulates features. (c) SPADE-Norm restores spatial cues "washed out" by standard normalization.

Latent diffusion models such as ControlNet Zhang et al. (2023) and T2I-Adapter Mou et al. (2024) often break fine connectivity when synthesizing from semantic maps: (i) the raw mask is downsampled by a factor of eight via convolution/interpolation, destroying small-scale connectivity and topology; and (ii) subsequent normalization (e.g., GroupNorm) spatially averages feature statistics, further washing out geometry Park et al. (2019); Lei et al. (2024b). We therefore introduce a **Topology-Consistent ControlNet (TC-ControlNet)** that preserves crack connectivity and topology at both the input and feature levels (Fig. 3(a)). The two key adaptations relative to vanilla ControlNet are (1) a *TopoDownsample* module and (2) topology-aware feature modulation via SPADE Norm.

(1) TopoDownsample module. Before the mask enters the latent UNet it must be reduced to $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$ of its original size. Naïve interpolation removes hairline cracks or merges adjacent regions. *TopoDownsample* performs this reduction while *exactly* preserving the connectivity and hole structure of every instance by casting downsampling as a small mixed-integer program (MIP) that assigns a component label to each low-resolution pixel. We maximize similarity to the original

mask while enforcing topology:

$$\begin{aligned}
&\max_{x} \sum_{m,i,j} w_m \, S_m(i,j) \, x_{m,i,j} \\
\text{s.t.} \ &\sum_{m} x_{m,i,j} = 1 \quad \text{(exclusivity)} \\
&\sum_{(i,j)\in\mathcal{R}_m} x_{m,i,j} \geq 1 \quad \text{(component survival)} \\
&x_{m,i,j} + x_{m,i+1,j+1} \leq 1 \quad \text{(avoid diagonal bridges)}
\end{aligned}$$

Here $x_{m,i,j} \in \{0,1\}$ indicates whether low-res pixel (i,j) is assigned to component m; w_m weights components (foreground $>$ background); and

$$S_m(i,j) = \underbrace{\frac{|\, m \cap \mathcal{N}(i,j)\,|}{|\mathcal{N}(i,j)|}}_{\text{local coverage}} + \lambda \underbrace{\frac{1}{1+\min_{p\in\partial m}\|p-(i,j)\|}}_{\text{boundary proximity}},$$

with $\mathcal{N}(i,j)$ a circular neighborhood (radius $2s_k$), ∂m the boundary of m, and λ=0.5. The first two constraints ensure every pixel takes exactly one label and no connected component disappears; the third keeps the background 4/8-connected at low resolution to prevent spurious holes. A small set of additional linear constraints (omitted here for brevity; see Appendix A.4) ensures boundary continuity so that foreground–background interfaces form a single closed loop. As a result, the downsampled masks preserve Euler characteristic, Betti numbers, and the region adjacency graph (RAG) of the original. Applying the MIP at three scales yields $\mathbf{M}^{(8)}, \mathbf{M}^{(16)}, \mathbf{M}^{(32)}$ fed to TC-ControlNet.

(2) Topology-aware feature modulation via SPADE Norm. To prevent normalization from blurring crack structure, every encoder block in the ControlNet branch replaces GroupNorm with SPADE Norm conditioned on topology-consistency masks (Fig. 3(b,c)). For an input feature tensor f and a mask $\mathbf{M}^{(s)}$ at scale $s \in \{8, 16, 32\}$, the layer computes

$$\mathrm{SPADE}(f, \mathbf{M}^{(s)}) = \gamma_s(\mathbf{M}^{(s)}) \, \tfrac{f-\mu(f)}{\sigma(f)} \; + \; \beta_s(\mathbf{M}^{(s)}), \tag{3}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are per-channel statistics, and the spatially varying scale and shift maps γ_s, β_s are produced by two 3×3 Conv–SiLU blocks. Feeding masks at three resolutions aligns the UNet's receptive field with expected crack widths and re-injects precise geometry that would otherwise be lost.

TC-ControlNet is trained with the same noise-prediction objective as vanilla ControlNet; only the mask embedder (some Conv layers) and the ControlNet branch are updated, while the Stable Diffusion backbone and the text encoder remain frozen.

4 Experiments

4.1 Experimental Setup

Datasets. All experiments are conducted on three high–quality crack datasets: *CrackInst1K*, *CRACK500* Shi et al. (2016), and *DeepCrack* Liu et al. (2019).

Evaluation protocol. Effectiveness is assessed from two complementary angles:

1. *Visual realism and consistency.* For every dataset we synthesise a set of image–mask pairs with CrackInstSynth+TC-ControlNet and compute
   - FID on RGB images (realism);
   - mIoU and absolute Betti errors β_0, β_1 between ground-truth masks and predictions of a pre-trained robust U-Net Lei et al. (2024b) (consistency).
2. *Downstream instance segmentation.* Each training set is enlarged five-fold using the full CrackInstSynth pipeline (only replaces TC-ControlNet with other generative models). A

Table 1: Visual realism and topology consistency on CRACK500, Deepcrack, and CrackInst1K. Best results are in **bold**.

Datasets	CRACK500				Deepcrack				CrackInst1K			
Methods	mIoU (↑)	FID (↓)	β_0 (↓)	β_1 (↓)	mIoU (↑)	FID (↓)	β_0 (↓)	β_1 (↓)	mIoU (↑)	FID (↓)	β_0 (↓)	β_1 (↓)
pix2pixHD	46.7	118.4	0.171	0.0120	48.2	148.9	55.00	37.45	54.2	142.3	52.04	32.44
SPADE	64.3	100.8	0.164	0.0115	63.4	137.5	50.84	33.35	74.0	125.4	39.21	26.15
SDM	62.1	98.3	0.140	0.0091	62.1	163.1	52.71	31.58	73.1	126.0	40.14	28.47
T2i-Adapter	52.4	94.5	0.151	0.0070	64.2	136.5	53.33	31.61	72.2	129.5	42.24	30.51
FreestyleNet	66.7	89.7	0.161	0.0088	65.3	122.1	48.51	31.46	74.8	113.1	36.14	26.86
ControlNet	73.4	90.3	0.168	0.0090	68.3	123.5	49.45	29.94	76.7	110.3	36.80	26.31
PLACE	71.3	88.6	0.124	0.0085	68.7	118.7	46.14	30.21	76.6	105.4	35.47	24.89
SCP-ControlNet	73.9	85.4	0.091	0.0073	67.6	117.6	44.77	31.22	76.0	103.8	34.28	22.95
TC-ControlNet (ours)	**75.2**	**82.3**	**0.071**	**0.0050**	**70.9**	**111.8**	**41.05**	**25.31**	**79.7**	**101.9**	**32.35**	**20.09**

standard Mask R-CNN He et al. (2017) detector is trained from scratch for 100 epochs (batch size ≥ 8) on the augmented data, and evaluated with $\text{mAP}_{50}^{\text{bbox}}$ and $\text{mAP}_{50}^{\text{seg}}$ on the held-out test split. For the semantic segmentation datasets *CRACK500* and *DeepCrack*, we consider there to be only one instance.

4.2 Experiment results and discussion

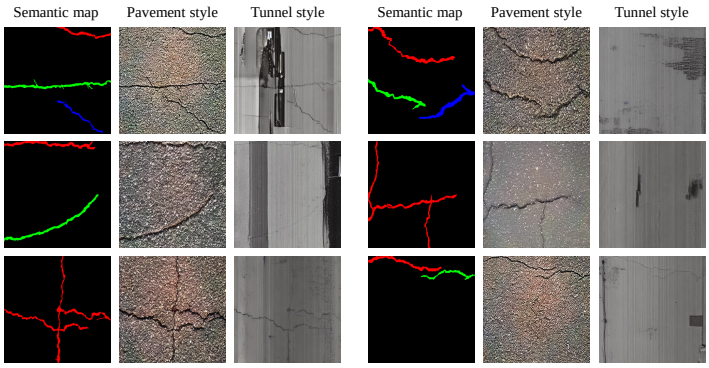

Figure 4: Qualitative results produced by **CrackInstSynth**. Each column group shows the input multi-instance semantic map (left) and the corresponding images generated in *pavement* and *tunnel* styles. In all cases the synthetic cracks remain topologically identical to the masks, even for hairline branches and intersections, while material appearance varies realistically across styles.

4.2.1 Evaluation of Visual Realism and Consistency

We compare TC-ControlNet with representative GANs (Pix2PixHD Wang et al. (2018), SPADE Park et al. (2019)), a pixel-space diffuser (SDM) Wang et al. (2022), and leading latent-space methods (T2i-Adapter Mou et al. (2024), FreestyleNet Xue et al. (2023), ControlNet Zhang et al. (2023), PLACE Lv et al. (2024), SCP-ControlNet Lei et al. (2024b)).

Table 1 shows that TC-ControlNet achieves the lowest FID and Betti errors and the highest mIoU on all three datasets. Relative to the strongest baseline (SCP-ControlNet) it cuts β_0 by 22%–30%, reduces β_1 by about 10%, and improves FID by 2–6, confirming that the TopoDownsample and SPADE-Norm mechanisms jointly preserve fine connectivity without sacrificing photorealism.

4.2.2 Evaluation of Downstream Segmentation Performance

Table 2 reports Mask R-CNN performance after augmenting each training set to five times its original size with different generators. CrackInstSynth paired with TC-ControlNet yields the highest $\text{mAP}_{50}^{\text{bbox}}$ and $\text{mAP}_{50}^{\text{seg}}$ on all three benchmarks, surpassing the strongest baseline (SCP-ControlNet) by up to $+1.5$ bbox AP and $+2.1$ mask AP. The gains over the "Original" rows confirm that the synthetic imagery is not only realistic but also *task-useful*, boosting instance detection and segmentation accuracy without additional manual labels.

Table 2: Downstream segmentation performance (Mask R-CNN) after 5× data augmentation on Deepcrack, CRACK500 and CrackInst1K. Best results are in **bold**.

Datasets	CRACK500		Deepcrack		CrackInst1K	
Methods	$mAP_{50}^{bbox}(\uparrow)$	$mAP_{50}^{seg}(\uparrow)$	$mAP_{50}^{bbox}(\uparrow)$	$mAP_{50}^{seg}(\uparrow)$	$mAP_{50}^{bbox}(\uparrow)$	$mAP_{50}^{seg}(\uparrow)$
Original	85.1	75.4	86.6	84.6	84.2	70.1
pix2pixHD	86.3	77.8	87.4	87.1	86.3	73.8
SPADE	87.9	79.6	88.9	88.0	88.1	76.5
SDM	87.4	79.1	88.5	87.6	87.6	76.0
T2i-Adapter	87.0	78.6	87.2	88.5	87.1	75.4
FreestyleNet	87.6	80.5	88.1	89.4	89.3	77.8
ControlNet	87.5	81.1	88.2	90.5	88.6	80.0
PLACE	87.0	81.6	88.8	91.0	89.4	79.2
SCP-ControlNet	87.4	82.5	89.0	90.9	89.1	81.1
TC-ControlNet	**88.7**	**84.2**	**89.7**	**92.2**	**91.2**	**83.3**

Table 3: Ablation study on **CrackInst1K** using downstream instance-segmentation metrics. Each variant removes or alters one component of CrackInstSynth.

ID	Method	mAP_{50}^{bbox} (↑)	mAP_{50}^{seg} (↑)
A0	Original training set (no aug.)	84.2	70.1
A1	No Region-level Placement (naïve paste)	86.4	75.4
A2	No Physics-driven Skeleton Growth	86.6	76.3
A3	Vanilla ControlNet (no topology branch)	88.6	80.0
A4	TC-ControlNet w/ TopoDownsample only	86.4	76.5
	TC-ControlNet w/ SPADE only	89.1	81.1
A5	**Full CrackInstSynth (ours)**	**91.2**	**83.3**

4.2.3 Visualization and Qualitative Analysis

Fig. 4 provides visual evidence that the proposed pipeline preserves crack geometry while offering flexible appearance control. TC-ControlNet renders photorealistic textures in two distinct styles, coarse asphalt and ribbed concrete tunnel.

4.2.4 Ablation Study

Table 3 isolates the contribution of each novel component on **CrackInst1K** dataset, reported with Mask R-CNN He et al. (2017) mAP after 5× augmentation:

A1 *No Region-level Placement.* Replacing RIP with naïve cut–paste lowers mask AP by 7.9. Overlapping or truncated instances therefore hurt detector training even when image realism is preserved. **A2** *No Physics-driven Skeleton Growth.* Skipping PSG removes the morphological diversity injected by the random walk, yielding a similar drop (-7.0 seg AP). Diversity in crack length and branching is thus essential for generalisation. **A3** *Vanilla ControlNet.* Using the standard latent UNet without our topology branch reduces bbox AP by 2.6 and seg AP by 3.3, the largest single loss. Preserving connectivity during Mask→Image synthesis is therefore critical. **A4** *TC-ControlNet variants.* Feeding only TopoDownsample masks (no SPADE) or only SPADE Norm (no TopoDownsample) recovers part of the gain, but neither matches the full model.

In sum, every module, RIP, PSG, and the dual innovations of TC-ControlNet, contributes measurably; removing any of them degrades instance segmentation, while the complete CrackInstSynth pipeline **A5** achieves the best accuracy (**+13.2** seg AP over the unaugmented baseline **A0**).

5 Conclusion

We introduced **CrackInst1K** (high-resolution tunnel-crack data with per-instance masks) and **CrackInstSynth** (a topology-aware augmentation pipeline combining RIP, PSG, and TPGM/TC-ControlNet), which synthesizes realistic, topology-consistent image–mask pairs and consistently boosts instance segmentation performance. We will release both resources to support future work in structural health monitoring.

REFERENCES

Nikhil Bansal, Vladan Urosevic, Simon Niklaus, et al. Panoptic diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Christian Benz and Volker Rodehorst. OmniCrack30k: A benchmark for crack segmentation and the reasonable effectiveness of transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3876–3886, 2024.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6154–6162, 2018.

Chia-Chia Chen and Chi-Han Peng. Topology-preserving downsampling of binary images. In *European Conference on Computer Vision*, pp. 416–431. Springer, 2024.

Zhuangzhuang Chen, Jin Zhang, Zhuonan Lai, Jie Chen, Zun Liu, and Jianqiang Li. Geometry-aware guided loss for deep crack recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4703–4712, 2022.

Zhuangzhuang Chen, Jin Zhang, Zhuonan Lai, Guanming Zhu, Zun Liu, Jie Chen, and Jianqiang Li. The devil is in the crack orientation: A new perspective for crack detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6653–6663, October 2023.

Zhuangzhuang Chen, Zhuonan Lai, Jie Chen, and Jianqiang Li. Mind marginal non-crack regions: Clustering-inspired representation learning for crack segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12698–12708, June 2024.

Zhuangzhuang Chen, Chengqi Xu, Tao Hu, Li Wang, Jie Chen, and Jianqiang Li. Decompose-compose feature augmentation for imbalanced crack recognition in industrial scenarios. *IEEE Transactions on Automation Science and Engineering*, 2025.

Bowen Cheng, Alexander Schwing, and Alexander Kirillov. 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.

Hsien-Hsiang Fang, Jian Ding, Lingxi Xie, et al. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, 2019.

Yuxin Fang, Shilin Yang, Jiewei Wang, Yuan Li, Juncheng Gao, Jian Deng, Jie Zhou, Xinlong Wang, and Chunhua Shen. Instances as queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6915–6925, 2021.

Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. Gurobi Optimization, LLC, 2024. URL `https://www.gurobi.com/documentation/`.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Hongwei Huang, Shuai Zhao, Dongming Zhang, and Jiayao Chen. Deep learning-based instance segmentation of cracks from shield tunnel lining images. *Structure and Infrastructure Engineering*, 18(2):183–196, 2022. doi: 10.1080/15732479.2020.1838559.

T Yung Kong and Azriel Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.

Nicholas Konz, Yuwen Chen, Haoyu Dong, and Maciej A Mazurowski. Anatomically-controllable medical image generation with segmentation-guided diffusion models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 88–98. Springer, 2024.

L. Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies—a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869–885, 1992. doi: 10.1109/34.161346.

Qin Lei, Jiang Zhong, Chen Wang, Yang Xia, and Yangmei Zhou. Dynamic thresholding for accurate crack segmentation using multi-objective optimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 389–404. Springer, 2023.

Qin Lei, Rui Yang, Jiang Zhong, Rongzhen Li, Muyang He, Mianxiong Dong, and Kaoru Ota. Expanding crack segmentation dataset with crack growth simulation and feature space diversity. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024a.

Qin Lei, Jiang Zhong, and Qizhu Dai. Enriching information and preserving semantic consistency in expanding curvilinear object segmentation datasets. In *European Conference on Computer Vision*, pp. 233–250. Springer, 2024b.

Qin Lei, Jiang Zhong, Chen Wang, and Xue Li. Integrating crack causal augmentation framework and dynamic binary threshold for imbalanced crack instance segmentation. *Expert Systems with Applications*, 240:122552, 2024c.

Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. CrackFormer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3783–3792, October 2021.

Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

Zhengyao Lv, Yuxiang Wei, Wangmeng Zuo, and Kwan-Yee K Wong. Place: Adaptive layout-semantic fusion for semantic image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9264–9274, 2024.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 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*, volume 38, pp. 4296–4304, 2024.

Fabio Panella, Aldo Lipani, and Jan Boehm. Semantic segmentation of cracks: Data challenges and architecture. *Automation in Construction*, 135:104110, 2022.

B. G. Pantoja-Rosero, Doruk Oner, Maciej Kozinski, Radhakrishna Achanta, Pascal Fua, Fernando Pérez-Cruz, and Karl Beyer. TOPO-Loss for continuity-preserving crack detection using deep learning. *Construction and Building Materials*, 344:128264, 2022. doi: 10.1016/j.conbuildmat.2022.128264.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, et al. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Amir Rezaie, Radhakrishna Achanta, Michele Godio, and Katrin Beyer. Comparison of crack segmentation using digital image correlation measurements and deep learning. *Construction and Building Materials*, 261:120474, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Azriel Rosenfeld. Digital topology. *The american Mathematical monthly*, 86(8):621–630, 1979.

Jiyuan Shi, Sal Saad Al Deen Taher, and Ji Dang. Comparison of semantic segmentation and instance segmentation based on pixel-level damage detection. In *AI·Data Science Proceedings*, pp. 46–53, 2021. doi: 10.11532/jsceiii.2.2_46.

Yuke Shi, Lili Wang, Bo Tian, and Yilong Yin. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016. doi: 10.1109/TITS.2016.2535360.

George Stockman and Linda G Shapiro. *Computer vision*. Prentice Hall PTR, 2001.

Xingyi Tian, Chunhua Shen, Hao Chen, and Tong He. Conditional convolutions for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 282–298, 2020.

Dapeng Wang, Jingchun Wang, Chengjie Rao, Xing Niu, and Qiang Xu. Tunnel lining crack expansion and maintenance strategy optimization considering train loads: A case study. *PLOS ONE*, 18(8):e0290533, 2023. doi: 10.1371/journal.pone.0290533.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, et al. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models. *arXiv preprint arXiv:2207.00050*, 2022.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 17727–17738, 2020.

Jiahao Xie, Wei Li, Xiangtai Li, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Mosaicfusion: Diffusion models as data augmenters for large vocabulary instance segmentation. *International Journal of Computer Vision*, 133(4):1456–1475, 2025.

Han Xue, Zhiwu Huang, Qianru Sun, Li Song, and Wenjun Zhang. Freestyle layout-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14256–14266, 2023.

Guo Yang, Tinghua Ai, and Yining Li. Crackgan: Pavement crack image synthesis using generative adversarial networks. *IEEE Transactions on Image Processing*, 29:9093–9108, 2020.

Fei Yu, Guanting Ye, Qing Jiang, Ka-Veng Yuen, Xun Chong, and Qiang Jin. Imaging-based instance segmentation of pavement cracks using an improved YOLOv8 network. *Structural Control and Health Monitoring*, 28(1):e3249, 2025. doi: 10.1002/stc.3249.

Qi Yuan, Yufeng Shi, and Mingyue Li. A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges. *Remote Sensing*, 16(16):2910, 2024. doi: 10.3390/rs16162910.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Mian Zhao, Xiangyang Xu, Xiaohua Bao, Xiangsheng Chen, and Hao Yang. An automated instance segmentation method for crack detection integrated with crackmover data augmentation. *Sensors*, 24(2):446, 2024a. doi: 10.3390/s24020446.

Yuanlin Zhao, Wei Li, Jiangang Ding, Yansong Wang, Lili Pei, and Aojia Tian. Crack instance segmentation using splittable transformer and position coordinates. *Automation in Construction*, 168:105838, 2024b.

Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3):1498–1512, 2018.

A APPENDIX

This appendix complements the main paper with six self-contained parts:

1. Section **Details of *CrackInst1K* Dataset** A.1 documents the new *CrackInst1K* dataset—its imaging pipeline, annotation protocol, statistics, example images, and anonymisation policy.
2. **Section More details about CrackInstSythn** A.2 give more Implementation details of CrackInstSythn.
3. Section **Rationale Behind *TC-ControlNet* Design** A.3 explains the design of *TC-ControlNet*, clarifying how TopoDownsample and SPADE jointly preserve crack topology and appearance.
4. Section **TopoDownsample: Formulation, Implementation & Theoretical Analysis** A.4 gives the full MIP formulation of TopoDownsample, solver details, qualitative comparisons, and a formal proof that the method preserves Betti numbers and the region-adjacency graph.
5. Section **Additional Details for Experiments** A.5 lists all hyper-parameters, hardware, prompts, detector settings, and evaluation metrics used in the experiments, followed by additional quantitative results—multi-detector gains, cross-dataset transfer, and sensitivity to the synthetic-to-real ratio.
6. Section **More visualizations** A.5.3 provides extra end-to-end visualisations, illustrating that the generated masks maintain perfect crack geometry across multiple rendering styles.

A.1 DETAILS OF *CrackInst1K* DATASET

A.1.1 SCOPE AND MOTIVATION

CrackInst1K is a public benchmark for instance-level crack segmentation in civil-infrastructure imagery. It contains 1025 tunnel-lining patches (1024×1024 pixels), released in COCO format. The dataset supplies scale-aware, topology-preserving annotations for algorithms that must separate neighbouring cracks and trace fine branches.

A.1.2 IMAGING AND PRE-PROCESSING PIPELINE

Images were captured in situ with a vehicle-mounted line-scan system. The rig maintains orthogonal viewing geometry and uniform illumination while travelling at approximately 3–5 km/h, producing raw stripes of 1000×7448 pixels that resolve cracks as thin as 0.29 mm. After acquisition, stripes were auto-stitched and frames containing visible cracks were retained. Each selected frame was cropped to square regions (800–3 000 pixels per side) and finally resized to 1024×1024 for release. Patches are split *per scene* to prevent leakage: 923 images for training and 102 for validation (90/10).

Fig. 5 gives a visual impression of the dataset, highlighting the thin, meandering geometry of cracks and the clutter commonly encountered inside tunnel environments.

A.1.3 ANNOTATION PROTOCOL

Annotation followed a four-stage procedure:

1. **Polygon tracing:** crack contours were digitised as dense polygons (mean 140 vertices).
2. **Instance labelling:** every polygon receives a unique identifier; intersecting cracks are traced separately.
3. **Automatic sanity checks:** scripts flag self-intersections, duplicate vertices or masks that leak outside the canvas.
4. **Double-blind review:** two independent annotators correct flagged masks; a third reviewer resolves conflicts.

The final JSON stores each polygon, its bounding box, skeleton length and the physical pixel size.

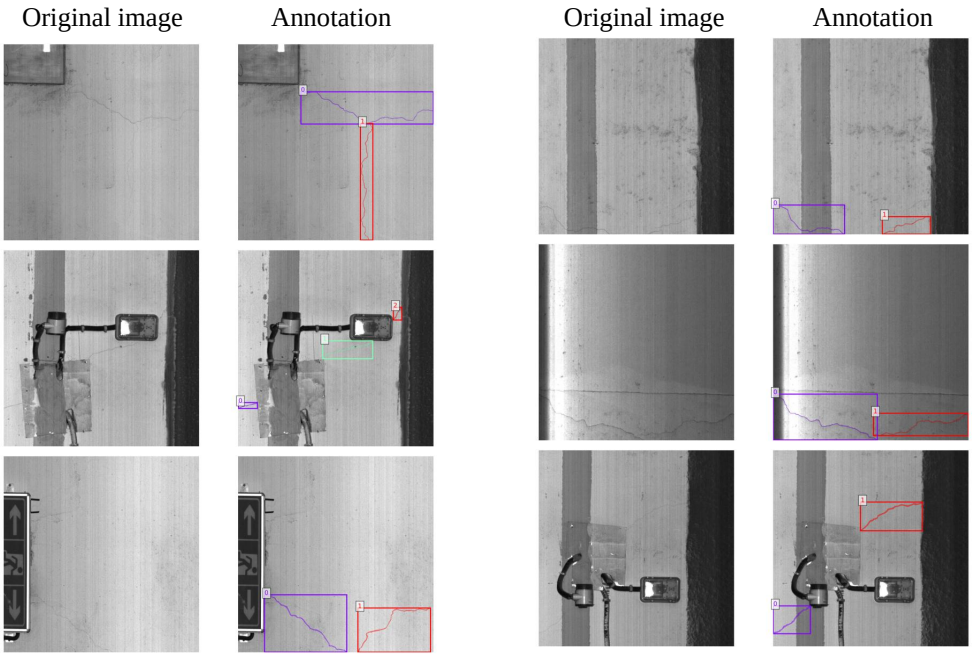

Figure 5: Representative samples from *CrackInst1K*. Each column pair shows the original image patch (left) and the corresponding instance annotation (right). Individual cracks are outlined with unique colours and numeric IDs, while background objects remain unlabelled to reflect real maintenance scenes.

A.1.4 Data Anonymisation and Availability

The raw imagery was collected over multiple years from several geographically distinct tunnels. During pre-processing, every exported 1024×1024 patch is a spatial mosaic drawn from different time stamps and camera poses. This strategy removes any location-specific patterns and prevents re-identification of the original infrastructure.

Availability – The dataset and scripts will be released upon acceptance to support reproducible research.

A.2 More details about CrackInstSythn

A.2.1 More detils about PSG

Algorithm 1 sketches the RandomWalk procedure.

Compared with pure geometric jittering, PSG implements a physics-based simulation of crack growth by carefully tuning the parameters of a bounded random walk, thereby increasing instance-level information content. In practice, PSG adds 35% new skeleton pixels per instance in average, providing diverse yet physically credible conditional masks for the subsequent Topology-Preserving Generation Module (TPGM).

A.2.2 More details about TPGM

TPGM $\mathcal{F}: \mathbf{S}_{\mathrm{PSG}} \longmapsto (\mathbf{M}_{\mathrm{WA}}, \mathbf{I})$ as much as possible meet the following conditions:

- *Intra-instance topology:* for every instance $c > 0$, the region $\{p : \mathbf{M}_{\mathrm{WA}}(p) = c\}$ forms a *single* 8-connected component (i.e., $\beta_0 = 1$, no holes).
- *Inter-instance separation:* $\forall p$ and $\forall c \neq d$, it never holds that $\mathbf{M}_{\mathrm{WA}}(p) = c \wedge \mathbf{M}_{\mathrm{WA}}(p) = d$ (regions are mutually exclusive).

Algorithm 1 Physics-driven random walk within a rescaled bbox

Input: skeleton $\mathbf{S}$, growth window $\hat{\mathcal{B}}$, parameters $(k, m, \mathit{min}, \mathit{max}, \ell, \theta)$
Initialise queue Q with k random start pixels on $\mathbf{S}$
$\mathbf{S}^{\text{new}} \leftarrow \mathbf{S}$
while $Q \neq \varnothing$ **do**
    Pop current point (x, y, d) where d stores the incoming direction
    if length$(\mathbf{S}^{\text{new}})$/ diag$(\hat{\mathcal{B}}) \geq m$ **or** steps $> \mathit{max}$ **then**
        continue
    end if
    Sample turning angle $\Delta\phi \sim \mathcal{U}(-\theta, \theta)$
    $d' \leftarrow d + \Delta\phi$; $(x', y') \leftarrow (x, y) + \ell(\cos d', \sin d')$
    if $(x', y') \in \hat{\mathcal{B}}$ **and** not occupied **then**
        Add (x', y') to $\mathbf{S}^{\text{new}}$ and push (x', y', d') to Q
    end if
end while
return $\mathbf{S}^{\text{new}}$

- *Pixel-wise consistency:* cracks in $\mathbf{I}$ correspond one-to-one with labels in $\mathbf{M}_{\text{WA}}$; in particular, the crack support in $\mathbf{I}$ exactly matches $\{p : \mathbf{M}_{\text{WA}}(p) \neq 0\}$, and for each $c > 0$ the rendered crack c coincides with $\{p : \mathbf{M}_{\text{WA}}(p) = c\}$.

A.2.3 More details about Stage 1: Skeleton→Mask Diffusion

Training pairs. We construct training pairs $(\mathbf{S}_{\text{sk}}, \mathbf{M}_{\text{gt}})$ from crack segmentation datasets by skeletonizing each ground-truth mask $\mathbf{M}_{\text{gt}}$ with Zhang–Suen thinning Lam et al. (1992) to obtain $\mathbf{S}_{\text{sk}}$. At inference time, $\mathbf{S}_{\text{PSG}}$ (from PSG) replaces $\mathbf{S}_{\text{sk}}$ as the conditioning input.

Hyperparameters. We follow the original SDM setup Wang et al. (2022); Lei et al. (2024a): cosine $\beta_{1:T}$ schedule, T=1000 training steps, DDIM 20 sampling steps, UNet depth 4 with 128 base channels, AdamW (learning rate 1×10^{-4}, weight decay 1×10^{-2}). We train for 150k iterations with batch size 16.

Sampling. Given a new skeleton mask, we sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and run the DDIM solver for 20 steps to obtain $\hat{\mathbf{M}}_{\text{WA}}$. A channelwise argmax yields the discrete width-aware map that feeds Stage 2.

A.3 Rationale Behind *TC-ControlNet* Design

Background. ControlNet Zhang et al. (2023) augments Stable Diffusion Rombach et al. (2022) with an extra condition c (for example, a segmentation map) to guide generation. The input mask $c \in \mathbb{R}^{H\times W\times 3}$ is first embedded by a small CNN, $h = E_c(c) \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 4}$, and added to the noisy latent z_t before entering the trainable branch of a U-Net copied from the frozen backbone.

Problem. GroupNorm layers inside the U-Net average spatial statistics and tend to erase fine semantic cues Park et al. (2019); Lei et al. (2024b). Moreover, naively downsampling thin-object masks to the latent resolution (1/8, 1/16, 1/32) breaks connectivity and alters topology, as shown in the main paper.

Solution. **TC-ControlNet** addresses both issues with two design choices:

1. **TopoDownsample**. The binary crack mask is reduced to three latent scales by solving a small mixed-integer program that preserves Euler characteristic and region-adjacency, avoiding the aliasing artefacts of convolutional or interpolation-based resizing.
2. **SPADE feature modulation.** Each encoder block replaces GroupNorm with SPADE Park et al. (2019), using the topology-safe masks from TopoDownsample as spatially varying scale and bias to reinject crack geometry lost during normalisation.

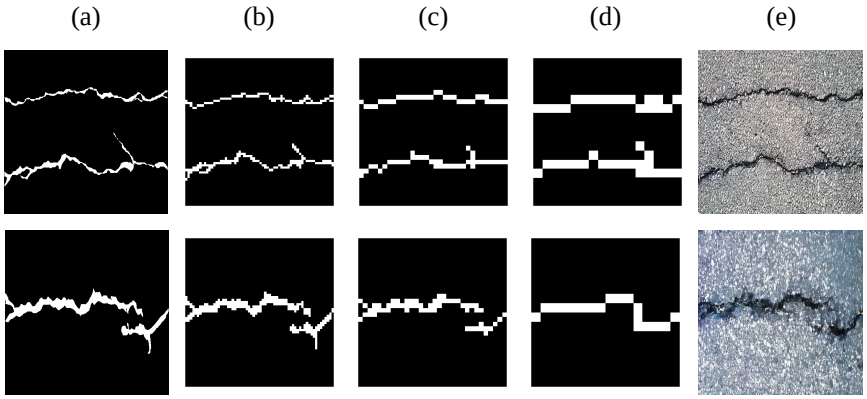

Figure 6: Effect of TopoDownsample and SPADE in TC-ControlNet. (a) Original 512×512 crack mask; (b)–(d) topology-preserving masks at resolutions 64, 32, and 16; (e) asphalt-style image synthesised by TC-ControlNet.

Illustration. Fig. 6 demonstrates the effect of the two modules on two representative crack masks. Column (a) shows the original 512×512 mask; columns (b)–(d) display the TopoDownsample outputs at resolutions 64, 32, and 16. Even at 1/32 resolution, the skeleton remains connected and free of spurious bridges. Column (e) gives the final photorealistic image generated by TC-ControlNet; the crack paths precisely match the conditioning masks, confirming that topology is retained through both the latent UNet and the RGB decoder.

Fallback strategy. If the mixed-integer solver fails to find a feasible assignment at a given scale (rare in practice), the algorithm falls back to bilinear interpolation for that scale only, ensuring continuity of the generation pipeline.

By combining topology-aware downsampling with spatially adaptive modulation, TC-ControlNet mitigates semantic dilution and delivers superior topological fidelity compared with vanilla ControlNet, as validated quantitatively and qualitatively in the main paper.

A.4 TopoDownsample: Formulation, Implementation & Theoretical Analysis

Simply resizing a thin–object mask from 512×512 to latent grids (64, 32, 16) with nearest, bicubic or pooling interpolation breaks connectivity and may introduce spurious holes. *TopoDownsample* addresses this by formulating down-sampling as a compact mixed-integer programme (MIP) whose feasible set contains *only* pixel assignments that preserve foreground–background topology Chen & Peng (2024).

1. Problem set-up

Let $c\in\{0,1\}^{H\times W}$ be the input binary mask and $c^{(k)}$ its coarse version at scale $k\in\{0,1,2\}$, height $H_k = H/2^{3-k}$. Each pixel of $c^{(k)}$ aggregates an $s_k\times s_k$ block of c, where $s_k = H/H_k$.

Decision variables.

- $x^{(k)}_{m,i,j}\in\{0,1\}$: macro-pixel (i,j) belongs to component m.
- $z^{(k)}_v\in\{0,1\}$: vertex v activates a valid corner configuration from the catalogue in Chen & Peng (2024).
- $l^{(k)}_v\in\{0,1\}$: unique terminal flag closing each boundary loop.

2. Objective

$$\max_{x,z,l} \sum_{m,i,j} w_m\, S_m(i,j)\, x^{(k)}_{m,i,j},$$

where $S_m(i,j)$ is the component–block overlap score and $w_m = 2$ for foreground, 1 for background Chen & Peng (2024).

3. CONSTRAINTS

1. **Exclusivity:** $\sum_m x^{(k)}_{m,i,j} = 1 \quad \forall (i,j)$.
2. **Component survival:** $\sum_{(i,j)\in\mathcal{R}_m} x^{(k)}_{m,i,j} \geq 1 \quad \forall m$.
3. **Anti-diagonal (background):** $x^{(k)}_{m,i,j} + x^{(k)}_{m,i+1,j+1} \leq 1$ for every background component m.
4. **Boundary continuity:** $z^{(k)}_v \Rightarrow \bigvee_{v'\in\mathcal{N}(v)} z^{(k)}_{v'}$.
5. **Loop closure:** $\sum_{v\in\mathcal{V}_b} l^{(k)}_v = 1$ for each boundary b.

Items 1–3 ensure a valid label map; 4–5 force every interface to form a single closed 8-connected curve, thereby preserving Euler characteristic $\chi = \beta_0 - \beta_1$ Kong & Rosenfeld (1989).

4. COMPONENT AND CORNER ENUMERATION

Foreground components are extracted with 8-connectivity, background with 4-connectivity—the standard "complementary" pairing that avoids paradoxes in digital topology Rosenfeld (1979). At each grid vertex we test the twelve corner templates of Chen & Peng (2024); invalid patterns are discarded, shrinking the MIP.

5. SOLVER DETAILS

The MIP is implemented in C++ and solved with Gurobi Gurobi Optimization, LLC (2024), while a Python port is provided for visualisation. A greedy warm start assigns each macro-pixel to the component covering the largest area. If a scale is infeasible (rare; $< 0.3\%$ at 16×16), TopoDownsample falls back to bilinear interpolation for that scale only.

6. QUALITATIVE COMPARISON

Fig. 7 contrasts TopoDownsample with five baselines (nearest, bicubic, pooling, ACN, dilation). Only our method preserves the crack's topology at 64, 32 and 16 pixels.

7. THEORETICAL ANALYSIS

We now prove that any feasible MIP solution preserves the Region Adjacency Graph (RAG) Stockman & Shapiro (2001) and the Betti numbers β_0 (components) and β_1 (holes).

Lemma 1. ***(Component preservation)*** *The down-sampled mask has exactly the same number of 8-connected foreground and 4-connected background components as the original; hence β_0 is unchanged.*

Proof. Component survival (Constraint 2) forbids disappearance. Anti-diagonal plus exclusivity (Constraints 1–3) forbid two distinct components from touching, preventing mergers. If an original component attempted to split, its boundary would fragment into two closed curves, violating the single-loop requirement (Constraint 5). Thus one-to-one correspondence of components holds. □

Lemma 2. ***(Hole preservation)*** *Every original hole persists in the down-sampled mask and no new hole is created; therefore β_1 is unchanged.*

Proof. A hole is a 4-connected background component fully enclosed by a foreground boundary. Lemma 1 guarantees the hole itself survives. Boundary continuity (Constraint 4) and unique-loop (Constraint 5) keep its enclosing Jordan curve intact, preventing the hole from leaking into exterior background. Because the MIP introduces no additional black–white adjacencies, no extra closed curve—and hence no extra hole—can arise. □

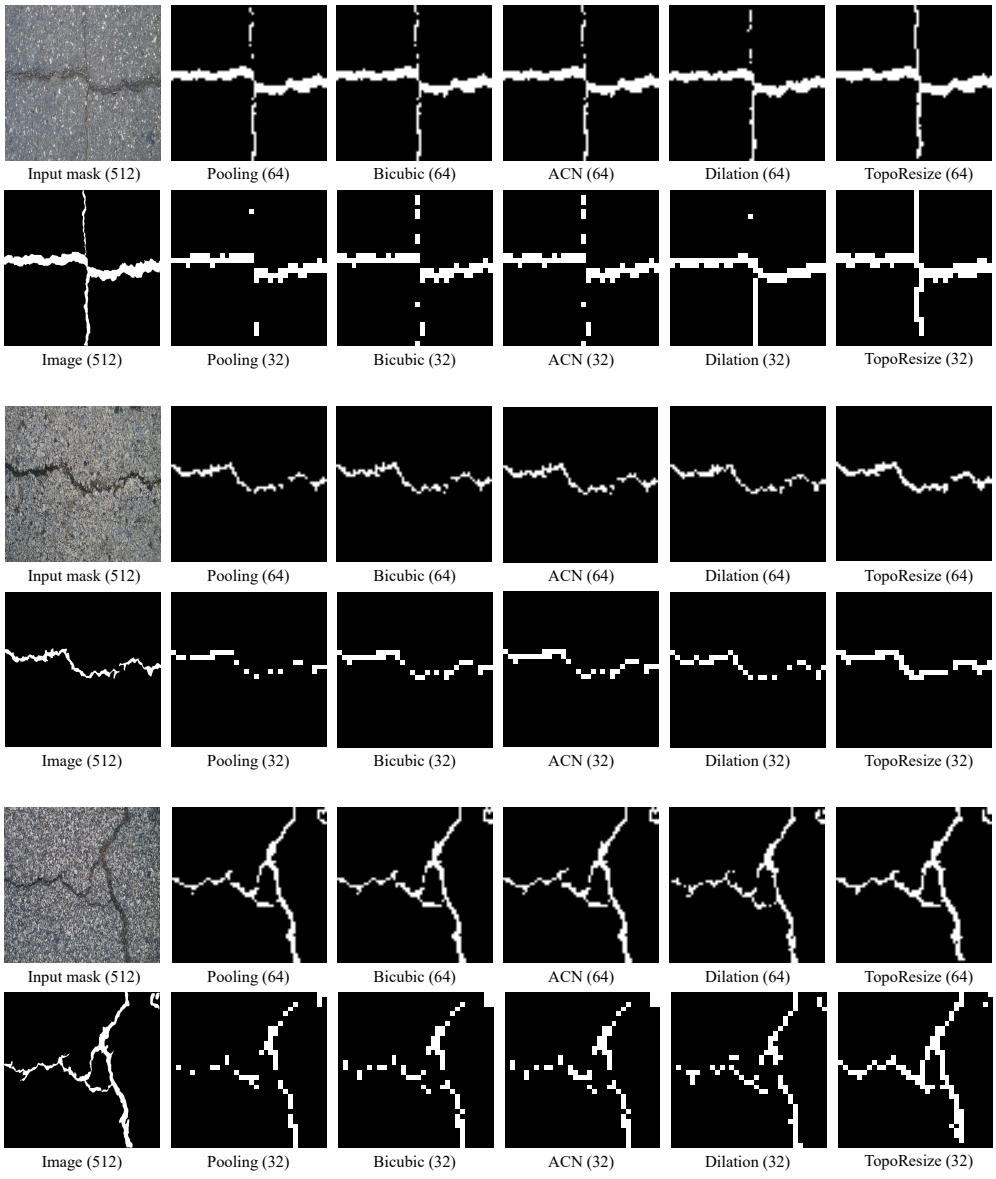

Figure 7: TopoDownsample versus conventional down-sampling methods on two crack masks.

Lemma 3. ***(RAG preservation)*** *The Region Adjacency Graph of the down-sampled mask is isomorphic to that of the input.*

Proof. By Lemma 1, nodes (regions) correspond one-to-one. For each original black–white pair, Constraint 5 instantiates exactly one closed boundary loop, producing the same edge in the output RAG. Pairs not adjacent originally remain separated by at least one pixel, and no new edge appears because no new boundary loop is allowed. □

Theorem 1. ***(Topology preservation)*** *Any feasible solution of the TopoDownsample MIP preserves β_0, β_1, and the entire RAG of the binary mask.*

Proof. Immediate from Lemmas 1, 2 and 3. □

A.5 Additional Details for Experiments

A.5.1 Experimental Setup

Hyper-parameter settings

All experiments were run on a workstation equipped with two NVIDIA RTX A6000 GPUs (48 GB each), an AMD EPYC 7513 CPU, and 256 GB RAM. Gurobi 11.0 is used for all MIP optimisations.

Unless stated otherwise, all hyper-parameters below are shared across *CrackInst1K*, *DeepCrack* and *CRACK500*.

- **Canvas tiling (RIP).** The 1024×1024 canvas is partitioned into four 512×512 quadrants; $n \sim \mathcal{U}\{1, 2, 3\}$ instances are sampled without replacement and pasted into randomly permuted quadrants. Source boxes are translated uniformly inside each quadrant, avoiding overlap.
- **BBox expansion for PSG.** For every seed instance the bounding box is isotropically enlarged by an independent scale factor $\alpha_x, \alpha_y \sim \mathcal{U}(1, 1+s)$ with $s = 0.6$. The enlarged window defines the admissible region for random walks.
- **Random-walk skeleton growth (PSG).** Number of starting points $k \sim \mathcal{U}\{0, \ldots, 4\}$; maximum crack-pixel ratio $m = 0.8$; step length 2 px; turning angle $\pm30^\circ$; step budget per start point $\texttt{min\_steps} = 30$, $\texttt{max\_steps} = 100$.
- **File resolution.** All intermediate masks are kept at 1024×1024; diffusion stages operate at 512×512 and outputs are up-sampled back to 1024 if needed using Lanczos.

Synthetic-data generation

- **SDM (Skeleton→Mask).** UNet depth 4, base channels 128; cosine β schedule, $T{=}1\,000$ training steps, DDIM 20 sampling steps; learning rate 1×10^{-4}, AdamW with weight decay 1×10^{-2}, batch size 16 for 150 k iterations.
- **TC-ControlNet (Mask→Image).** Frozen SD 1.5 backbone, ControlNet channel multiplier 0.5, TopoDownsample at scales $64/32/16$, SPADE injection at every encoder block, classifier-free guidance scale 7.5, DDIM 20 inference steps.
- **Prompt settings.** For the Crack500 dataset, the prompt settings from COSTG Lei et al. (2024b) were used. For the Deepcrack and CrackInst1K datasets, the prompt templates are as follows: *An image of cracks in a tunnel lining (road pavement); CrackInst1K dataset (Deepcrack dataset); there are(is) k cracks(s) in this image.* Here $k = \{1, 2, 3\}$ is the number of crack instances randomly placed during RIP.
- **Augmentation budget.** Each training split is expanded to exactly $5\times$ its original size (Table 2, main paper). Synthetic images are saved at 1024×1024, then centre-cropped to 1024×1024 before training.

Downstream instance segmentation

The goal of this experiment group is to measure how much **CrackInstSynth** augments improve *downstream* crack instance segmentation. Unless noted otherwise, every detector is trained *twice*:

(i) on the original real-image training split and (ii) on the augmented set (real + synthetic at $5\times$ scale). All hyper-parameters below are kept identical between the two runs so that any accuracy difference can be attributed solely to the synthetic data.

- **Mask R-CNN He et al. (2017).** ResNet-50-FPN backbone initialised from COCO; SGD (lr 0.02, momentum 0.9, wd 1×10^{-4}); linear warm-up 1k iters, step drops at epochs 60 and 80; 100 epochs, global batch size 8 (CrackInst1K) or 12 (DeepCrack / CRACK500); data aug. = random flip (p 0.5) + scale jitter $[0.8, 1.2]$; evaluation with COCO AP at IoU 0.50 using `coco_eval.py` in Detectron2.
- **Cascade Mask R-CNN Cai & Vasconcelos (2018).** R50-FPN, three-stage cascade; other settings identical to Mask R-CNN; configuration follows the official Detectron2 recipe.
- **Mask2Former Cheng et al. (2022).** Swin-L backbone, 2× LR schedule (100 epochs on our datasets), AdamW optimiser with parameters from the original paper; all other hyper-parameters unchanged.
- **CondInst Tian et al. (2020).** R50-FPN; trained with default 3× schedule in Detectron2 but capped at 100 epochs for parity.
- **SOLOv2 Wang et al. (2020).** ResNet-101-FPN; we adopt the authors' public configuration from MMDetection 3.3; total epochs 100.
- **QueryInst Fang et al. (2021).** Swin-T backbone; default learning-rate schedule from the paper; batch size 8 due to GPU memory.

Loss weights, anchor settings, and post-processing remain exactly as in the respective reference implementations; no detector-specific tuning is performed.

Image-quality evaluation

- **FID.** Computed on 10k CrackInstSynth images *vs.* the entire real training split of the same dataset; Inception-V3 *pool3* features, TORCH-FID.
- **mIoU and Betti errors.** Each *instance* mask is collapsed to a *binary* crack-vs-background map before evaluation so that topology metrics reflect true foreground connectivity, independent of instance IDs. A robust U-Net Lei et al. (2024b), predicts binary masks for 2k synthetic images; results are compared to ground truth to obtain mIoU as well as absolute Betti-number errors ($|\Delta\beta_0|$, $|\Delta\beta_1|$). Connected-component analysis uses SCIKIT-IMAGE `measure.label` with 8-connectivity for foreground and 4-connectivity for background, matching the TopoDownsample convention.

A.5.2 More Experimental Results

More detectors performance on CrackInstSynth

Table 4 reports the $\text{mAP}^{\text{seg}}_{50}$ achieved by six detectors on three benchmarks, *with* and *without* CrackInstSynth augmentation. All models gain accuracy, with TC-ControlNet data giving the largest boost on the most data-hungry detector (Mask R-CNN).

Table 4: Segmentation mAP50 before and after adding CrackInstSynth training data.

Detector	CrackInst1K		DeepCrack		CRACK500	
	Base	+Synth	Base	+Synth	Base	+Synth
Mask R-CNN	70.1	83.3	84.6	92.2	75.4	84.2
Cascade Mask R-CNN	71.5	82.0	85.3	91.0	76.0	83.6
Mask2Former	78.0	86.1	89.4	94.0	80.2	87.1
CondInst	69.4	78.4	83.2	88.5	73.0	80.6
SOLOv2	65.0	73.2	79.8	85.1	68.7	75.8
QueryInst	79.3	86.3	78.7	93.7	80.1	85.0

Cross-dataset generalisation

To evaluate domain robustness, we adopt a leave-one-dataset-out protocol: the detector is trained on a single *source* set (with or without CrackInstSynth) and tested *zero-shot* on the remaining two *target* sets. Table 5 shows that synthetic training data consistently improve segmentation accuracy by **6–9** across all six source–target pairs.

Table 5: Cross-dataset generalisation of Mask R-CNN between datasets. $\text{mAP}_{50}^{\text{seg}}$.

Source	Target	Base	+Synth
CrackInst1K	DeepCrack	60.2	**67.5**
	CRACK500	58.9	**65.1**
DeepCrack	CrackInst1K	62.1	**69.3**
	CRACK500	61.0	**68.2**
CRACK500	CrackInst1K	58.0	**64.8**
	DeepCrack	59.1	**66.4**

Synthetic samples derived from the *source* domain thus transfer positively, even without any access to *target* images, confirming CrackInstSynth's value for real-world deployment.

Sensitivity to the synthetic-to-real ratio

Fig. 8 shows how Mask R-CNN $\text{mAP}_{50}^{\text{seg}}$ evolves when the synthetic-to-real ratio increases from 1:1 to 50:1. The trend is consistent across all three benchmarks: accuracy climbs rapidly up to a 5:1 ratio and saturates thereafter. Consequently, we fix $5\times$ synthetic augmentation for all main-paper experiments.

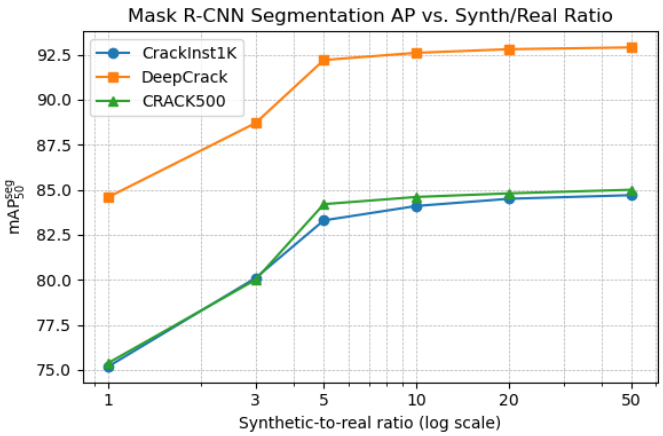

Figure 8: Mask R-CNN segmentation AP versus the amount of synthetic data, plotted on a log-scaled x-axis.

A.5.3 More visualizations

Fig. 9 gives six additional skeleton→mask→image examples. The first two columns illustrate how the physics-driven skeleton growth and Skeleton2Mask diffusion jointly recover realistic widths and branches. Columns (c) and (d) demonstrate the flexibility of TC-ControlNet: with an identical mask conditioning, prompting for *tunnel lining* versus *asphalt pavement* yields visually plausible textures for two very different materials, yet the coloured crack instances remain pixel-aligned. Such label-faithful, multi-style synthesis is central to the domain transfer experiments reported in Section **Cross-dataset generalisation**.

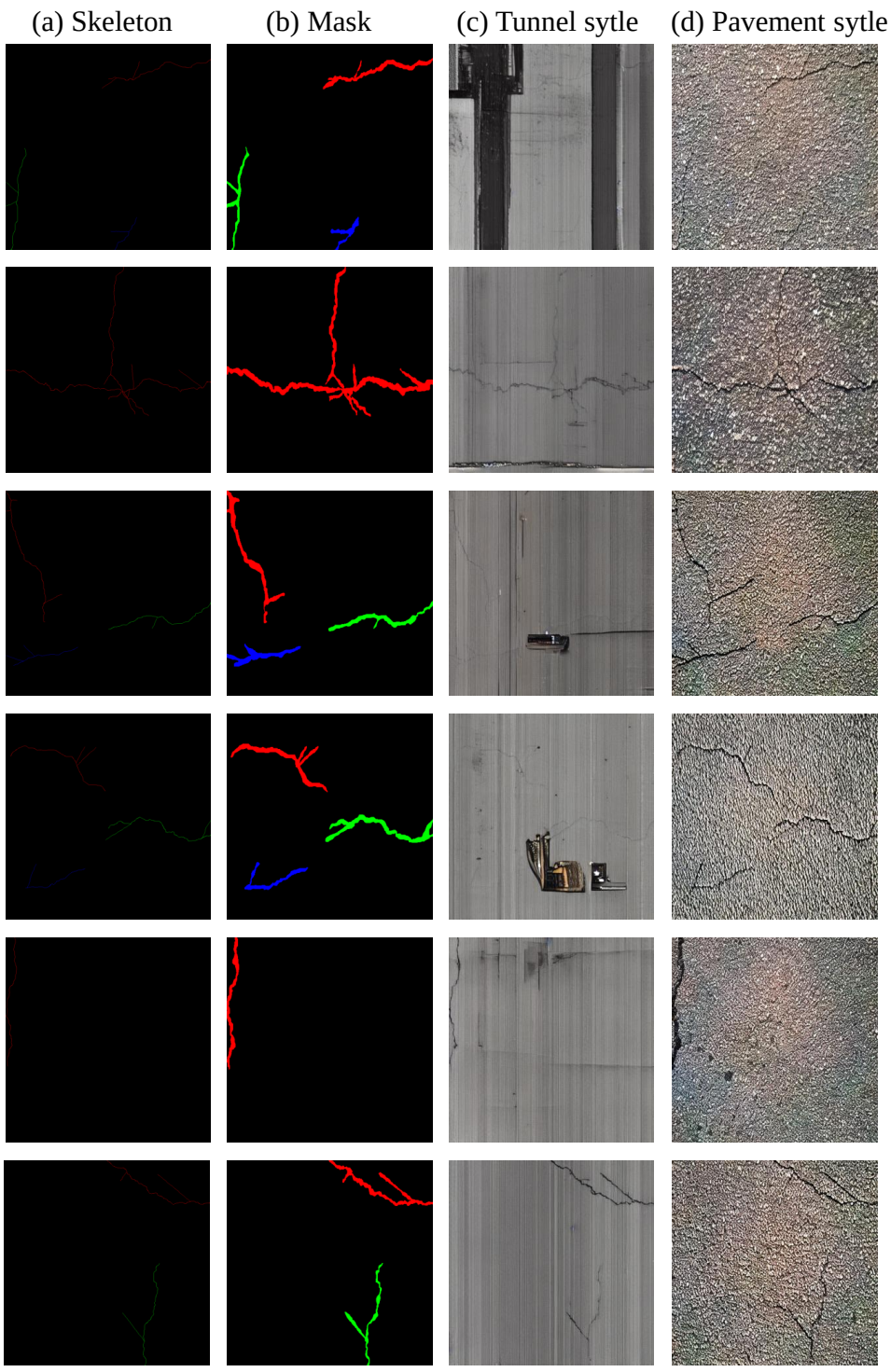

Figure 9: Additional end-to-end results of **CrackInstSynth**. (a) input multi-instance skeletons after physics-driven growth; (b) width-aware masks generated by the Skeleton→Mask diffusion; (c) tunnel-style renderings produced by TC-ControlNet; (d) the same masks rendered in a pavement style by simply switching the text prompt. Across all six rows the crack geometry (colours denote instance IDs) is preserved exactly, confirming topology fidelity, while background appearance adapts convincingly to the requested domain.