

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 CRACKINSTSYNTH: TOPOLOGY-AWARE GENERATIVE DATA-AUGMENTATION FRAMEWORK FOR CRACK IN- STANCE 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.

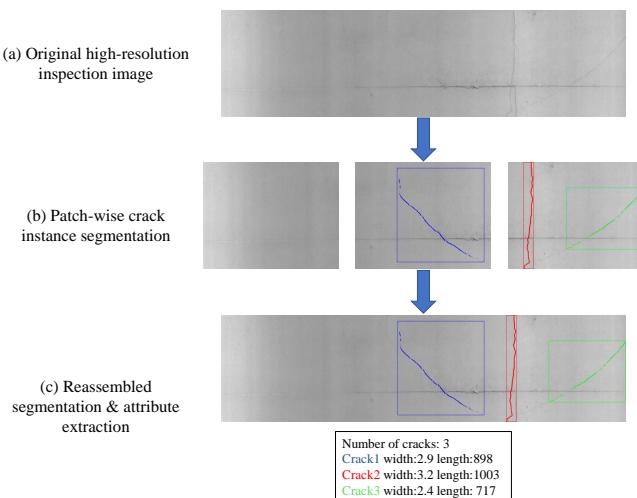
1 INTRODUCTION

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

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070



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

077
078
079 et al. (2016) contain only a few hundred images, and collecting additional data is costly: a single
 080 tunnel survey typically yields fewer than 500 usable frames, while pixel-level annotation by safety
 081 experts can take several minutes per image Panella et al. (2022); Liu et al. (2019); Rezaie et al.
 082 (2020). The shortage is even more acute for instance-level labels: no open dataset currently provides
 083 pixel-accurate instance IDs (unique labels for each crack), and most studies rely on binary masks
 084 or sparsely annotated subsets Zhao et al. (2024a). This gap hampers automatic structural health
 085 monitoring and highlights the need for cost-effective methods to enlarge corpora for crack instance
 086 segmentation.

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

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

108 modules, CrackInstSynth generates diverse, label-aligned samples that augment real data and enable
 109 large-scale training of crack instance segmentation models.
 110

111 We assess CrackInstSynth on CrackInst1K as well as the public DeepCrack Liu et al. (2019) and
 112 CRACK500 Shi et al. (2016) benchmarks. The evaluation covers both the visual fidelity of the syn-
 113 synthetic images and the performance gains of the synthetic image-mask pairs provide when training
 114 state-of-the-art instance segmentation models. In every case, the augmented data yields clear im-
 115 provements over baseline training sets, confirming the practical value of the proposed framework
 116 for crack analysis. We will release CrackInst1K and CrackInstSynth to support future research.
 117

In summary, this paper makes the following contributions:

1. **CrackInst1K**: a publicly available dataset of 1025 tunnel-lining images (1024×1024) with
 119 pixel-accurate per-instance masks. Images are de-identified via tile-mosaic preprocessing;
 120 the set focuses on hairline and branched/intersecting cracks, forming a fine-grained bench-
 121 mark.
2. **CrackInstSynth**: a topology-aware generative augmentation framework that produces di-
 123 verse, label-aligned image-mask pairs without extra labeling, integrating RIP, PSG, and
 124 TPGM with a two-stage diffusion pipeline (skeleton \rightarrow mask, mask \rightarrow image).
 125

126 2 RELATED WORK

127 2.1 CRACK INSTANCE SEGMENTATION

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

135 More recent work adapts generic instance frameworks to the thin-object regime of cracks. Mask R-
 136 CNN He et al. (2017) pipelines augmented with morphological closing reconnect fragmented masks
 137 and improve accuracy on tunnel linings Huang et al. (2022). Orientation-aware detectors represent a
 138 curved crack as a sequence of rotated segments to separate crossings and branches Chen et al. (2023).
 139 Lightweight one-stage networks enhanced with multi-head and triplet attention achieve real-time
 140 instance segmentation while boosting recall for fine structures Yu et al. (2025). Other approaches
 141 advance per-crack labeling but still struggle with continuity and ambiguous boundaries Zhao et al.
 142 (2024b); Lei et al. (2024c).

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

150 2.2 GENERATIVE DATA AUGMENTATION

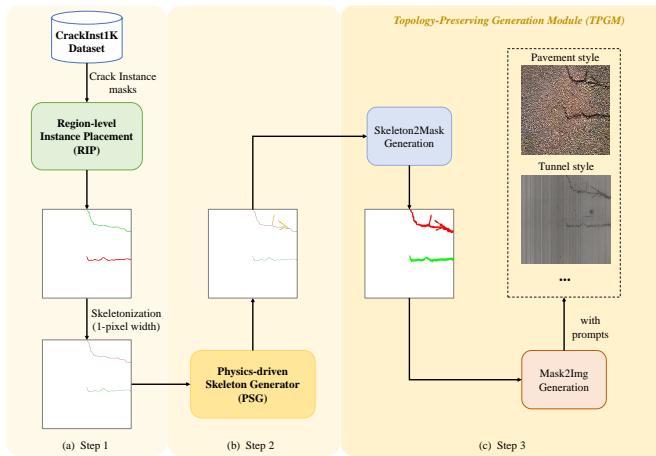
152 Conditional GANs were the first practical engines for paired data synthesis. pix2pixHD and SPADE
 153 translate coarse semantic masks into photorealistic surfaces and have been adapted to crack and road-
 154 defect imagery Wang et al. (2018); Park et al. (2019). Such GAN-based pipelines boost realism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

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

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

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

172 3 METHODOLOGY



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

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

208 3.1 REGION-LEVEL INSTANCE PLACEMENT

210 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
 211 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
 212 $H = W = 1024$, RIP synthesises a layout with $n \sim \mathcal{U}\{1, 2, 3\}$ instances.

213 The canvas is partitioned into four non-overlapping quadrants $\mathcal{R} = \{R^1, R^2, R^3, R^4\}$, each of size
 214 512×512 . RIP selects n distinct regions $\{R^{\pi(1)}, \dots, R^{\pi(n)}\}$ by a random permutation π , then
 215 places the n sampled masks after an i.i.d. translation

216
217
218
219
220
221

$$\mathbf{t}_i \sim \mathcal{U}([0, w_i^{\max}] \times [0, h_i^{\max}]),$$

$$w_i^{\max} = w_{R^{\pi(i)}} - w_j,$$

$$h_i^{\max} = h_{R^{\pi(i)}} - h_j.$$
(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}_{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 $\sim 1000, \text{px} \approx 1, \text{m}$, a 1024×1024 canvas (about $1, \text{m}^2$) 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 $\mathbf{S}_i = \text{THIN}(\tilde{\mathbf{M}}_i)$ using Zhang–Suen thinning Lam 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 ($k \sim \mathcal{U}\{1, \dots, M_k\}$), 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:

$$\mathbf{S}_i^{\text{new}} = \text{RANDOMWALK}(\mathbf{S}_i, \hat{\mathcal{B}}_i, k, m, \text{max}, \ell, \theta),$$

where *max* are the maximum step counts, ℓ is the step length in pixels (default 2) and θ bounds the turning angle ($\pm 30^\circ$). The walk is confined to $\hat{\mathcal{B}}_i$ to avoid inter-instance overlap.

See Appendix A.3.1 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* $\mathbf{S}_{\text{PSG}} \in \{0, \dots, n\}^{H \times W}$ by assigning value i to pixels on $\mathbf{S}_i^{\text{new}}$ and 0 to background. Each pixel of \mathbf{S}_{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* $\mathbf{M}_{\text{WA}} \in \{0, \dots, n\}^{H \times W}$ that expands each skeleton to its physical width, and a photorealistic image \mathbf{I} , through two diffusion stages.

Formally, TPGM learns

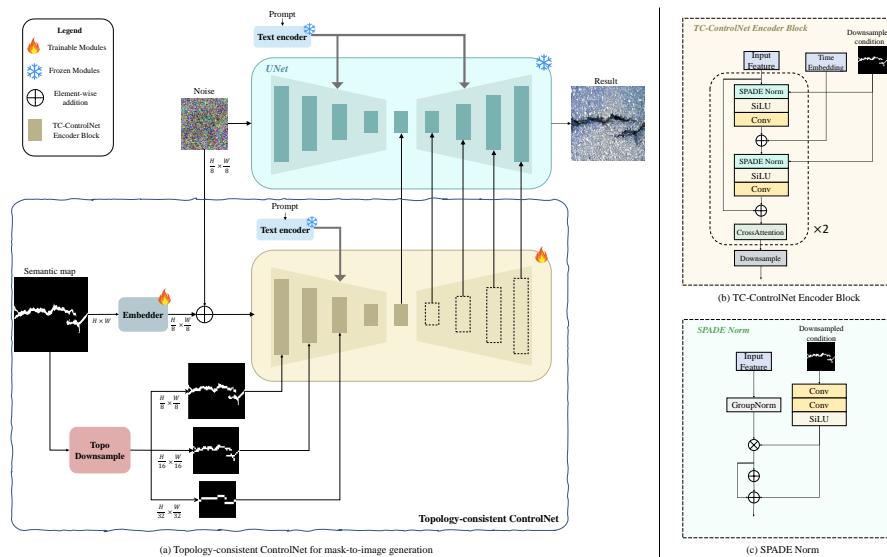
$$\mathcal{F} : \mathbf{S}_{\text{PSG}} \longmapsto (\mathbf{M}_{\text{WA}}, \mathbf{I}). \quad (2)$$

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

270 3.3.1 STAGE 1: SKELETON→MASK DIFFUSION
271

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

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

282
283 3.3.2 STAGE 2: MASK→IMAGE DIFFUSION
284
285

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

311
312 Latent diffusion models such as ControlNet Zhang et al. (2023) and T2I-Adapter Mou et al. (2024)
313 often break fine connectivity when synthesizing from semantic maps: (i) the raw mask is down-
314 sampled by a factor of eight via convolution/interpolation, destroying small-scale connectivity and
315 topology; and (ii) subsequent normalization (e.g., GroupNorm) spatially averages feature statistics,
316 further washing out geometry Park et al. (2019); Lei et al. (2024b). We therefore introduce a
317 **Topology-Consistent ControlNet (TC-ControlNet)** that preserves crack connectivity and topology
318 at both the input and feature levels (Fig. 3(a)). The two key adaptations relative to vanilla ControlNet
319 are (1) a *TopoDownsample* module and (2) topology-aware feature modulation via SPADE Norm.

320 **(1) TopoDownsample module.** Before the mask enters the latent UNet it must be reduced to $\frac{1}{8}$,
321 $\frac{1}{16}$, and $\frac{1}{32}$ of its original size. Naïve interpolation removes hairline cracks or merges adjacent
322 regions. *TopoDownsample* performs this reduction while *exactly* preserving the connectivity and
323 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

324 mask while enforcing topology:
 325

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

330 Here $x_{m,i,j} \in \{0, 1\}$ indicates whether low-res pixel (i, j) is assigned to component m ; w_m weights
 331 components (foreground > background); and
 332

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

333 with $\mathcal{N}(i,j)$ a circular neighborhood (radius $2s_k$), ∂m the boundary of m , and $\lambda=0.5$. The first
 334 two constraints ensure every pixel takes exactly one label and no connected component disappears;
 335 the third keeps the background 4/8-connected at low resolution to prevent spurious holes. A small
 336 set of additional linear constraints (omitted here for brevity; see Appendix A.5) ensures boundary
 337 continuity so that foreground–background interfaces form a single closed loop. As a result, the
 338 downsampled masks preserve Euler characteristic, Betti numbers, and the region adjacency graph
 339 (RAG) of the original. Applying the MIP at three scales yields $\mathbf{M}^{(8)}, \mathbf{M}^{(16)}, \mathbf{M}^{(32)}$ fed to TC-
 340 ControlNet.

341 **(2) Topology-aware feature modulation via SPADE Norm.** To prevent normalization from
 342 blurring crack structure, every encoder block in the ControlNet branch replaces GroupNorm with
 343 SPADE Norm conditioned on topology-consistency masks (Fig. 3(b,c)). For an input feature tensor
 344 f and a mask $\mathbf{M}^{(s)}$ at scale $s \in \{8, 16, 32\}$, the layer computes
 345

$$\text{SPADE}(f, \mathbf{M}^{(s)}) = \gamma_s(\mathbf{M}^{(s)}) \frac{f - \mu(f)}{\sigma(f)} + \beta_s(\mathbf{M}^{(s)}), \quad (3)$$

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

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

353 4 EXPERIMENTS

354 4.1 EXPERIMENTAL SETUP

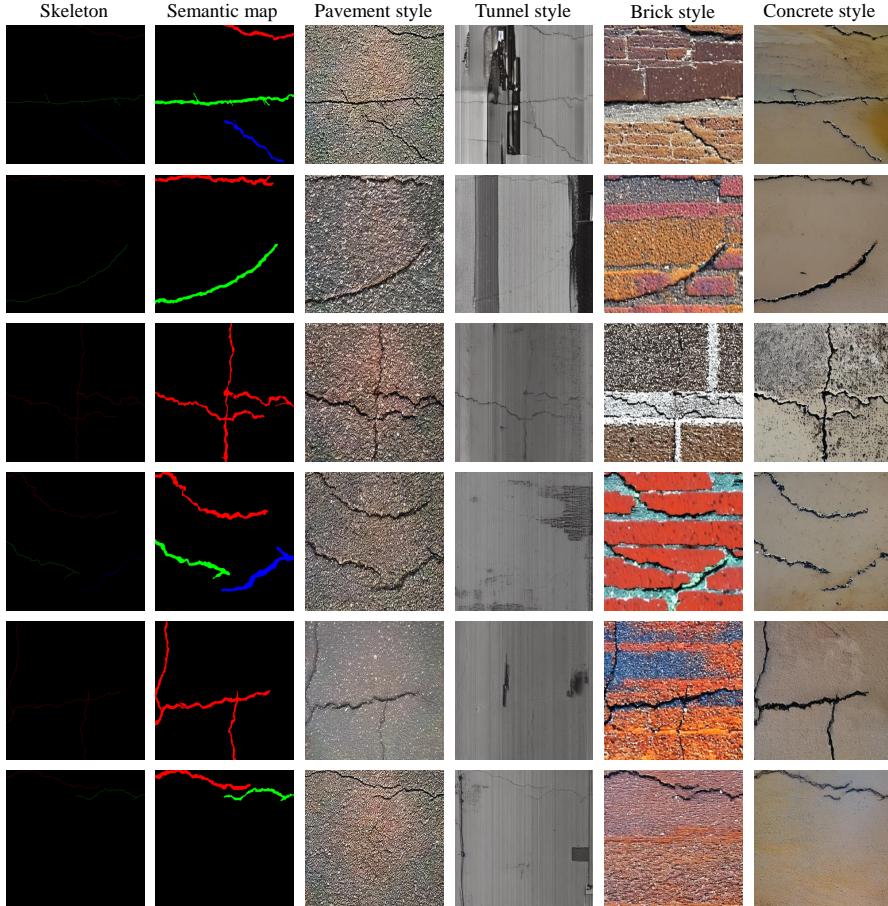
355 **Datasets.** All experiments are conducted on three high-quality crack datasets: *CrackInst1K*,
 356 *CRACK500* Shi et al. (2016), and *DeepCrack* Liu et al. (2019).

357 **Evaluation protocol.** Effectiveness is assessed from two complementary angles:

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

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

383 4.2 EXPERIMENT RESULTS AND DISCUSSION



416 Figure 4: Qualitative results produced by **CrackInstSynth**. Columns (left→right): *Skeleton*, *Semantic map*, *Pavement style*, *Tunnel style*, *Brick style*, *Concrete style*. For each row we keep the
 417 mask (instance IDs shown by colours) fixed and only change the text prompt; TC-ControlNet trans-
 418 fers the background/material appearance while preserving per-instance topology and width, demon-
 419 strating label-faithful, multi-style rendering. Best viewed in colour and with zoom.

423 4.2.1 EVALUATION OF VISUAL REALISM AND CONSISTENCY

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

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

432
 433 **Table 1: Visual realism and topology consistency on CRACK500, Deepcrack, and CrackInst1K.**
 434 **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 (↓)
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

443
 444 **Table 2: Downstream segmentation performance (Mask R-CNN) after 5× data augmentation on**
 445 **Deepcrack, CRACK500 and CrackInst1K. Best results are in bold.**

Datasets	CRACK500			Deepcrack			CrackInst1K		
	Methods	mAP ₅₀ ^{bbox} (↑)	mAP ₅₀ ^{seg} (↑)	mAP ₅₀ ^{bbox} (↑)	mAP ₅₀ ^{seg} (↑)	mAP ₅₀ ^{bbox} (↑)	mAP ₅₀ ^{seg} (↑)	mAP ₅₀ ^{bbox} (↑)	mAP ₅₀ ^{seg} (↑)
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			

4.2.2 EVALUATION OF DOWNSTREAM SEGMENTATION PERFORMANCE

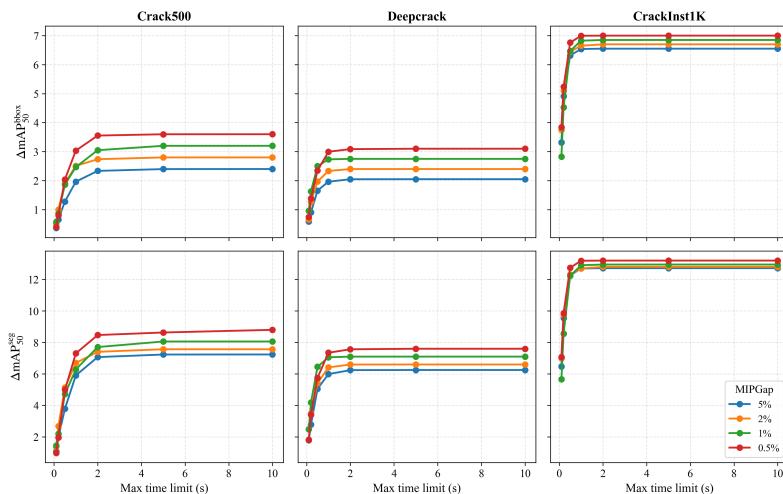
461 Table 2 reports Mask R-CNN performance after augmenting each training set to five times its original size with different generators. CrackInstSynth paired with TC-ControlNet yields the highest
 462 mAP₅₀^{bbox} and mAP₅₀^{seg} on all three benchmarks, surpassing the strongest baseline (SCP-ControlNet)
 463 by up to +1.5 bbox AP and +2.1 mask AP. The gains over the “Original” rows confirm that the synthetic
 464 imagery is not only realistic but also *task-useful*, boosting instance detection and segmentation
 465 accuracy without additional manual labels.

4.2.3 VISUALIZATION AND QUALITATIVE ANALYSIS

471 Fig. 4 provides visual evidence that the proposed pipeline preserves crack geometry while offering
 472 flexible appearance control. TC-ControlNet renders photorealistic textures in various distinct styles,
 473 coarse asphalt, ribbed concrete tunnel brick masonry facade and concrete structural surface.

474
 475 **Table 3: Ablation study on CrackInst1K using downstream instance-segmentation metrics. Each**
 476 **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

486 4.2.4 ABLATION STUDY
487488 Table 3 isolates the contribution of each novel component on **CrackInst1K** dataset, reported with
489 Mask R-CNN He et al. (2017) mAP after $5\times$ augmentation:490 **A1 No Region-level Placement.** Replacing RIP with naïve cut–paste lowers mask AP by 7.9. Over-
491 lapping or truncated instances therefore hurt detector training even when image realism is preserved.
492 **A2 No Physics-driven Skeleton Growth.** Skipping PSG removes the morphological diversity injected
493 by the random walk, yielding a similar drop (-7.0 seg AP). Diversity in crack length and branching
494 is thus essential for generalisation. **A3 Vanilla ControlNet.** Using the standard latent UNet without
495 our topology branch reduces bbox AP by 2.6 and seg AP by 3.3, the largest single loss. Preserv-
496 ing connectivity during Mask→Image synthesis is therefore critical. **A4 TC-ControlNet variants.**
497 Feeding only TopoDownsample masks (no SPADE) or only SPADE Norm (no TopoDownsample)
498 recovers part of the gain, but neither matches the full model.499 In sum, every module, RIP, PSG, and the dual innovations of TC-ControlNet, contributes measur-
500 ably; removing any of them degrades instance segmentation, while the complete CrackInstSynth
501 pipeline **A5** achieves the best accuracy (**+13.2** seg AP over the unaugmented baseline **A0**).502 503 4.2.5 RUNTIME-ACCURACY TRADE-OFF OF TOPODOWNSAMPLE
504505 We study the effect of the MIP solver tolerance on accuracy and latency for TopoDownsample. Fig. 5
506 (one representative dataset) plots accuracy gain vs. max time limit under $\text{MIPGap} \in \{5, 2, 1, 0.5\}\%$.
507 The curves show monotonic but saturating improvements: tightening the gap yields higher accuracy
508 with diminishing returns. In practice, we select the tightest feasible gap under a given time budget;
509 if the budget is exceeded, we *fallback* to bilinear downsampling to guarantee responsiveness.526 Figure 5: Runtime–accuracy trade-off of TopoDownsample. Accuracy gain vs. max time limit under
527 different MIPGap settings on a representative dataset.
528529 Across datasets and metrics, a moderate tolerance (MIPGap 1–2%) within a 1–2 s budget captures
530 almost all of the attainable gains, while tighter settings incur disproportionate latency for marginal
531 improvements.
532533 5 CONCLUSION
534535 We introduced **CrackInst1K** (high-resolution tunnel-crack data with per-instance masks) and
536 **CrackInstSynth** (a topology-aware augmentation pipeline combining RIP, PSG, and TPGM/TC-
537 ControlNet), which synthesizes realistic, topology-consistent image–mask pairs and consistently
538 boosts instance segmentation performance. We will release both resources to support future work in
539 structural health monitoring.

540 REFERENCES
541

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

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

547

548 Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In
549 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.
550 6154–6162, 2018.

551 Saúl Cano-Ortiz, Eugenio Sainz-Ortiz, Lara Lloret Iglesias, Pablo Martínez Ruiz del Árbol, and
552 Daniel Castro-Fresno. Enhancing pavement crack segmentation via semantic diffusion synthesis
553 model for strategic road assessment. *Results in Engineering*, 23:102745, 2024.

554

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

557

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

561 Zhuangzhuang Chen, Jin Zhang, Zhuonan Lai, Guanming Zhu, Zun Liu, Jie Chen, and Jianqiang
562 Li. The devil is in the crack orientation: A new perspective for crack detection. In *Proceedings*
563 *of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6653–6663, October
564 2023.

565

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

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

573 Bowen Cheng, Alexander Schwing, and Alexander Kirillov. Masked-attention mask transformer for
574 universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision*
575 *and Pattern Recognition (CVPR)*, pp. 1290–1299, 2022.

576

577 Gozde Merve Demirci, Jiaqi Yang, Hyun Sung Song, Chao Chen, Wei-Chi Wu, and Chia-Ling
578 Tsai. Topology-aware conditional latent diffusion for multi-view fundus image synthesis. In
579 *Proceedings of the ACM/IEEE International Conference on Connected Health: Applications,*
580 *Systems and Engineering Technologies*, pp. 453–457, 2025.

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

583

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

587

588 Saumya Gupta, Dimitris Samaras, and Chao Chen. Topodiffusionnet: A topology-aware diffusion
589 model. *arXiv preprint arXiv:2410.16646*, 2024.

590 Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. Gurobi Optimization, LLC, 2024.
591 URL <https://www.gurobi.com/documentation/>.

592

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

594 Hongwei Huang, Shuai Zhao, Dongming Zhang, and Jiayao Chen. Deep learning-based instance
 595 segmentation of cracks from shield tunnel lining images. *Structure and Infrastructure Engineering*,
 596 18(2):183–196, 2022. doi: 10.1080/15732479.2020.1838559.

597

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

600 Nicholas Konz, Yuwen Chen, Haoyu Dong, and Maciej A Mazurowski. Anatomically-controllable
 601 medical image generation with segmentation-guided diffusion models. In *International Conference
 602 on Medical Image Computing and Computer-Assisted Intervention*, pp. 88–98. Springer,
 603 2024.

604

605 L. Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies—a comprehensive survey.
 606 *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869–885, 1992. doi:
 607 10.1109/34.161346.

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

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

614

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

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

621

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

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

627

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

631 Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan.
 632 T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion
 633 models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–
 634 4304, 2024.

635

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

638

639 B. G. Pantoja-Rosero, Doruk Oner, Maciej Kozinski, Radhakrishna Achanta, Pascal Fua, Fernando
 640 Pérez-Cruz, and Karl Beyer. TOPO-Loss for continuity-preserving crack detection using deep
 641 learning. *Construction and Building Materials*, 344:128264, 2022. doi: 10.1016/j.conbuildmat.
 642 2022.128264.

643

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

647

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

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

651

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

653

654 Jiyuan Shi, Sal Saad Al Deen Taher, and Ji Dang. Comparison of semantic segmentation and instance
 655 segmentation based on pixel-level damage detection. In *AI-Data Science Proceedings*, pp. 46–53,
 656 2021. doi: 10.11532/jscieiii.2.2.46.

657

658 Yuke Shi, Lili Wang, Bo Tian, and Yilong Yin. Automatic road crack detection using random
 659 structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445,
 660 2016. doi: 10.1109/TITS.2016.2535360.

661

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

663

664 Xingyi Tian, Chunhua Shen, Hao Chen, and Tong He. Conditional convolutions for instance seg-
 665 mentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 282–
 666 298, 2020.

667

668 Dapeng Wang, Jingchun Wang, Chengjie Rao, Xing Niu, and Qiang Xu. Tunnel lining crack expan-
 669 sion and maintenance strategy optimization considering train loads: A case study. *PLOS ONE*,
 670 18(8):e0290533, 2023. doi: 10.1371/journal.pone.0290533.

671

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

675

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

678

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

682

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

686

687 Meilong Xu, Saumya Gupta, Xiaoling Hu, Chen Li, Shahira Abousamra, Dimitris Samaras, Prateek
 688 Prasanna, and Chao Chen. Topocellgen: Generating histopathology cell topology with a diffusion
 689 model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20979–
 20989, 2025.

690

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

694

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

697

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

700

701 Qi Yuan, Yufeng Shi, and Mingyue Li. A review of computer vision-based crack detection methods
 702 in civil infrastructure: Progress and challenges. *Remote Sensing*, 16(16):2910, 2024. doi: 10.
 3390/rs16162910.

702

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

702 Mian Zhao, Xiangyang Xu, Xiaohua Bao, Xiangsheng Chen, and Hao Yang. An automated instance
 703 segmentation method for crack detection integrated with crackmover data augmentation. *Sensors*,
 704 24(2):446, 2024a. doi: 10.3390/s24020446.

705 Yuanlin Zhao, Wei Li, Jiangang Ding, Yansong Wang, Lili Pei, and Aoja Tian. Crack instance
 706 segmentation using splittable transformer and position coordinates. *Automation in Construction*,
 707 168:105838, 2024b.

708 Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian
 709 Zhang, Xuetao Wei, and Yuxuan Liang. Controltraj: Controllable trajectory generation with
 710 topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on
 711 Knowledge Discovery and Data Mining*, pp. 4676–4687, 2024.

712 Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learn-
 713 ing hierarchical convolutional features for crack detection. *IEEE Transactions on Image Process-
 714 ing*, 28(3):1498–1512, 2018.

717 A APPENDIX

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

- 721 1. Section **Additional related work** A.1 show additional related work, including the recent
 722 works about topology-aware diffusion for curvilinear structures.
- 724 2. Section **Details of CrackInst1K Dataset** A.2 documents the new *CrackInst1K* dataset—
 725 its imaging pipeline, annotation protocol, statistics, example images, and anonymisation
 726 policy.
- 727 3. Section **More details about CrackInstSynth** A.3 give more Implementation details of
 728 CrackInstSynth.
- 729 4. Section **Rationale Behind TC-ControlNet Design** A.4 explains the design of *TC-
 730 ControlNet*, clarifying how TopoDownsample and SPADE jointly preserve crack topology
 731 and appearance.
- 732 5. Section **TopoDownsample: Formulation, Implementation & Theoretical Analysis** A.5
 733 gives the full MIP formulation of TopoDownsample, solver details, qualitative compar-
 734 isons, and a formal proof that the method preserves Betti numbers and the region-adjacency
 735 graph.
- 736 6. Section **Additional Details for Experiments** A.6 lists all hyper-parameters, hardware,
 737 prompts, detector settings, and evaluation metrics used in the experiments, followed by
 738 additional quantitative results.
- 739 7. Section **More visualizations** A.6.3 provides extra end-to-end visualisations, illustrating
 740 that the generated masks maintain perfect crack geometry across multiple rendering styles.

741 A.1 ADDITIONAL RELATED WORK

743 A.1.1 TOPOLOGY-AWARE DIFFUSION FOR CURVILINEAR STRUCTURES

745 Recent works inject explicit topological objectives into diffusion. TopoDiffusionNet Gupta et al.
 746 (2024) and TopoCellGen Xu et al. (2025) incorporate persistent-homology (PH) constraints to
 747 guide denoising toward target Betti profiles, improving topological faithfulness beyond appear-
 748 ance similarity. For linear networks, ControlTraj Zhu et al. (2024) enforces path-level constraints
 749 under diffusion to maintain global connectivity and branching structure. In medical imaging, a
 750 topology-aware conditional LDM preserves vascular connectivity and branching via PH-guided
 751 losses across views Demirci et al. (2025). Closer to cracks, semantic diffusion-based pavement
 752 synthesis improves realism and segmentation but does not explicitly regulate connectivity or non-
 753 self-intersection during generation Cano-Ortiz et al. (2024).

754 Unlike PH-loss-based approaches (e.g., TopoDiffusionNet) that condition on multi-object topology,
 755 we follow the semantics-to-image paradigm and focus on single-crack instance topology consis-
 756 tency. Concretely, we adopt a two-stage design that couples a physics-guided skeleton generator

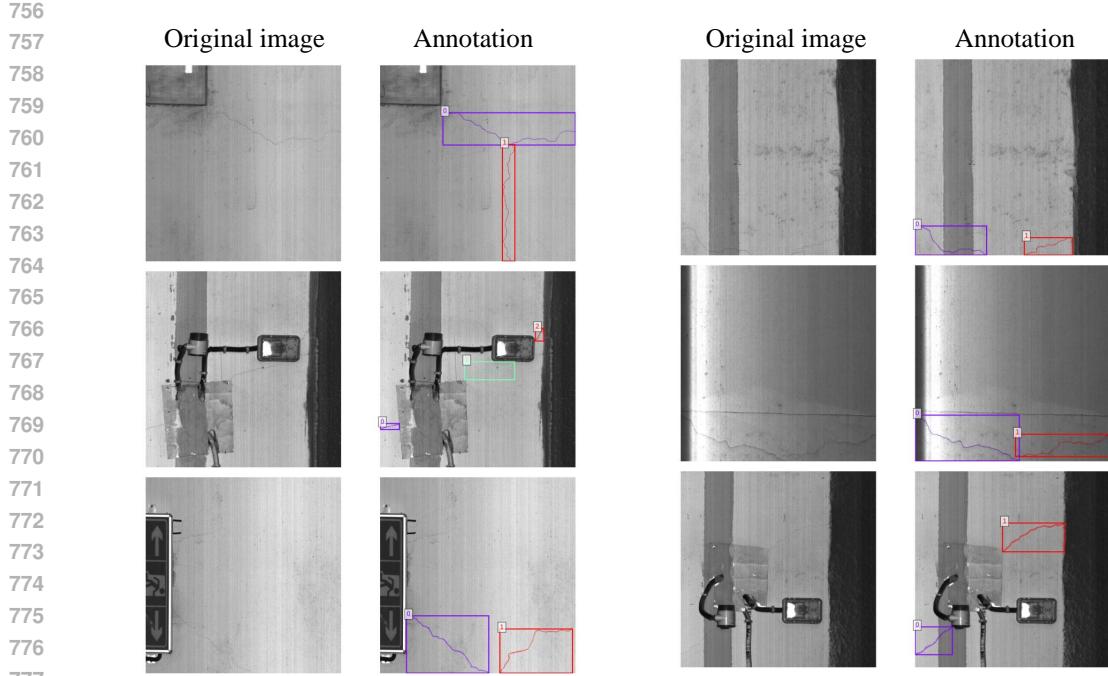


Figure 6: Representative samples from *CrackInst1K*. Each column pair shows the original image patch (left) and the corresponding instance annotation (right). Individual cracks are outlined with unique colours and numeric IDs, while background objects remain unlabelled to reflect real maintenance scenes.

(PSG) with a topology-preserving conditioning module, injecting crack-specific priors to yield more informative synthetic samples for downstream instance analysis.

A.2 DETAILS OF *CrackInst1K* DATASET

A.2.1 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.2.2 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. 6 gives a visual impression of the dataset, highlighting the thin, meandering geometry of cracks and the clutter commonly encountered inside tunnel environments.

A.2.3 ANNOTATION PROTOCOL

Annotation followed a four-stage procedure:

1. **Polygon tracing:** crack contours were digitised as dense polygons (mean 140 vertices).

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

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.

A.2.4 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.3 MORE DETAILS ABOUT CRACKINSTSYTHN

A.3.1 MORE DETILS ABOUT PSG

Algorithm 1 sketches the RandomWalk procedure.

Algorithm 1 Physics-driven random walk within a rescaled bbox

```

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

```

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.3.2 MORE DETAILS ABOUT TPGM

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

- *Intra-instance topology:* for every instance $c > 0$, the region $\{p : \mathbf{M}_{\text{WA}}(p) = c\}$ 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 $\mathbf{M}_{\text{WA}}(p) = c \wedge \mathbf{M}_{\text{WA}}(p) = d$ (regions are mutually exclusive).

864 • *Pixel-wise consistency*: cracks in \mathbf{I} correspond one-to-one with labels in \mathbf{M}_{WA} ; in partic-
 865 ular, the crack support in \mathbf{I} exactly matches $\{p : \mathbf{M}_{\text{WA}}(p) \neq 0\}$, and for each $c > 0$ the
 866 rendered crack c coincides with $\{p : \mathbf{M}_{\text{WA}}(p) = c\}$.
 867

868 A.3.3 MORE DETAILS ABOUT STAGE 1: SKELETON→MASK DIFFUSION
 869

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

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

877 *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
 878 steps to obtain $\hat{\mathbf{M}}_{\text{WA}}$. A channelwise argmax yields the discrete width-aware map that feeds Stage 2.
 879

880 A.4 RATIONALE BEHIND *TC-ControlNet* DESIGN
 881

882 **Background.** ControlNet Zhang et al. (2023) augments Stable Diffusion Rombach et al. (2022)
 883 with an extra condition c (for example, a segmentation map) to guide generation. The input mask
 884 $c \in \mathbb{R}^{H \times W \times 3}$ 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
 885 latent z_t before entering the trainable branch of a U-Net copied from the frozen backbone.
 886

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

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

892 1. **TopoDownsample**. The binary crack mask is reduced to three latent scales by solving
 893 a small mixed-integer program that preserves Euler characteristic and region-adjacency,
 894 avoiding the aliasing artefacts of convolutional or interpolation-based resizing.
 895 2. **SPADE feature modulation**. Each encoder block replaces GroupNorm with SPADE Park
 896 et al. (2019), using the topology-safe masks from TopoDownsample as spatially varying
 897 scale and bias to reinject crack geometry lost during normalisation.
 898

899 **Illustration.** Fig. 7 demonstrates the effect of the two modules on two representative crack masks.
 900 Column (a) shows the original 512×512 mask; columns (b)–(d) display the TopoDownsample out-
 901 puts at resolutions 64, 32, and 16. Even at 1/32 resolution, the skeleton remains connected and free
 902 of spurious bridges. Column (e) gives the final photorealistic image generated by TC-ControlNet;
 903 the crack paths precisely match the conditioning masks, confirming that topology is retained through
 904 both the latent UNet and the RGB decoder.
 905

906 **Fallback strategy.** If the mixed-integer solver fails to find a feasible assignment at a given scale
 907 (rare in practice), the algorithm falls back to bilinear interpolation for that scale only, ensuring
 908 continuity of the generation pipeline.
 909

910 By combining topology-aware downsampling with spatially adaptive modulation, TC-ControlNet
 911 mitigates semantic dilution and delivers superior topological fidelity compared with vanilla Control-
 912 Net, as validated quantitatively and qualitatively in the main paper.
 913

914 A.5 TOPODOWNSAMPLE: FORMULATION, IMPLEMENTATION & THEORETICAL ANALYSIS
 915

916 Simply resizing a thin-object mask from 512×512 to latent grids (64, 32, 16) with nearest, bicubic
 917 or pooling interpolation breaks connectivity and may introduce spurious holes. *TopoDownsample*

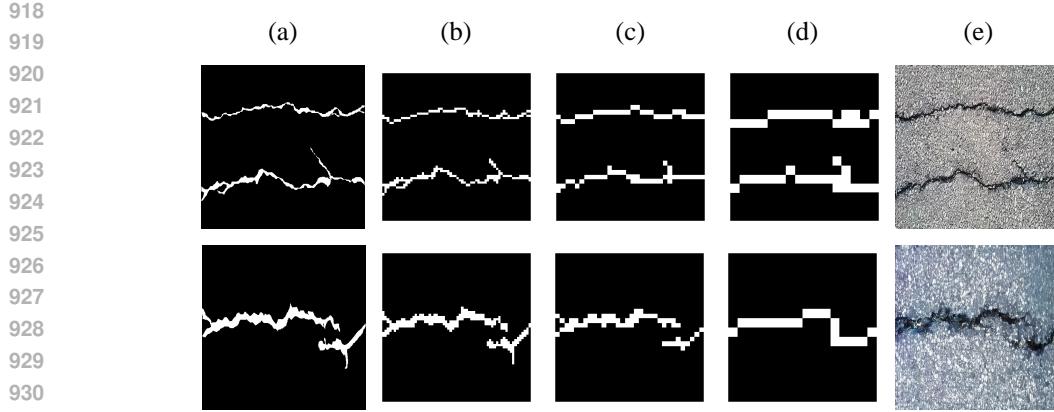


Figure 7: Effect of TopoDownsample and SPADE in TC-ControlNet. (a) Original 512×512 crack mask; (b)–(d) topology-preserving masks at resolutions 64 , 32 , and 16 ; (e) asphalt-style image synthesised by TC-ControlNet.

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.

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

2. OBJECTIVE

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

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

972 4. COMPONENT AND CORNER ENUMERATION
973974 Foreground components are extracted with 8-connectivity, background with 4-connectivity—the
975 standard “complementary” pairing that avoids paradoxes in digital topology Rosenfeld (1979). At
976 each grid vertex we test the twelve corner templates of Chen & Peng (2024); invalid patterns are
977 discarded, shrinking the MIP.978
979 5. SOLVER DETAILS
980981 The MIP is implemented in C++ and solved with Gurobi Gurobi Optimization, LLC (2024), while a
982 Python port is provided for visualisation. A greedy warm start assigns each macro-pixel to the com-
983 ponent covering the largest area. If a scale is infeasible (rare; $< 0.3\%$ at 16×16), TopoDownsample
984 falls back to bilinear interpolation for that scale only.985
986 6. QUALITATIVE COMPARISON
987988 Fig. 8 contrasts TopoDownsample with five baselines (nearest, bicubic, pooling, ACN, dilation).
989 Only our method preserves the crack’s topology at 64, 32 and 16 pixels.990
991 7. THEORETICAL ANALYSIS
992993 We now prove that any feasible MIP solution preserves the Region Adjacency Graph (RAG) Stock-
994 man & Shapiro (2001) and the Betti numbers β_0 (components) and β_1 (holes).995 **Lemma 1. (Component preservation)** *The down-sampled mask has exactly the same number of
996 8-connected foreground and 4-connected background components as the original; hence β_0 is un-
997 changed.*998 *Proof.* Component survival (Constraint 2) forbids disappearance. Anti-diagonal plus exclusivity
999 (Constraints 1–3) forbid two distinct components from touching, preventing mergers. If an original
1000 component attempted to split, its boundary would fragment into two closed curves, violating the
1001 single-loop requirement (Constraint 5). Thus one-to-one correspondence of components holds. \square 1002 **Lemma 2. (Hole preservation)** *Every original hole persists in the down-sampled mask and no new
1003 hole is created; therefore β_1 is unchanged.*1004 *Proof.* A hole is a 4-connected background component fully enclosed by a foreground boundary.
1005 Lemma 1 guarantees the hole itself survives. Boundary continuity (Constraint 4) and unique-loop
1006 (Constraint 5) keep its enclosing Jordan curve intact, preventing the hole from leaking into exterior
1007 background. Because the MIP introduces no additional black–white adjacencies, no extra closed
1008 curve—and hence no extra hole—can arise. \square 1009 **Lemma 3. (RAG preservation)** *The Region Adjacency Graph of the down-sampled mask is isomor-
1010 phic to that of the input.*1011 *Proof.* By Lemma 1, nodes (regions) correspond one-to-one. For each original black–white pair,
1012 Constraint 5 instantiates exactly one closed boundary loop, producing the same edge in the output
1013 RAG. Pairs not adjacent originally remain separated by at least one pixel, and no new edge appears
1014 because no new boundary loop is allowed. \square 1015 **Theorem 1. (Topology preservation)** *Any feasible solution of the TopoDownsample MIP preserves
1016 β_0 , β_1 , and the entire RAG of the binary mask.*1017 *Proof.* Immediate from Lemmas 1, 2 and 3. \square

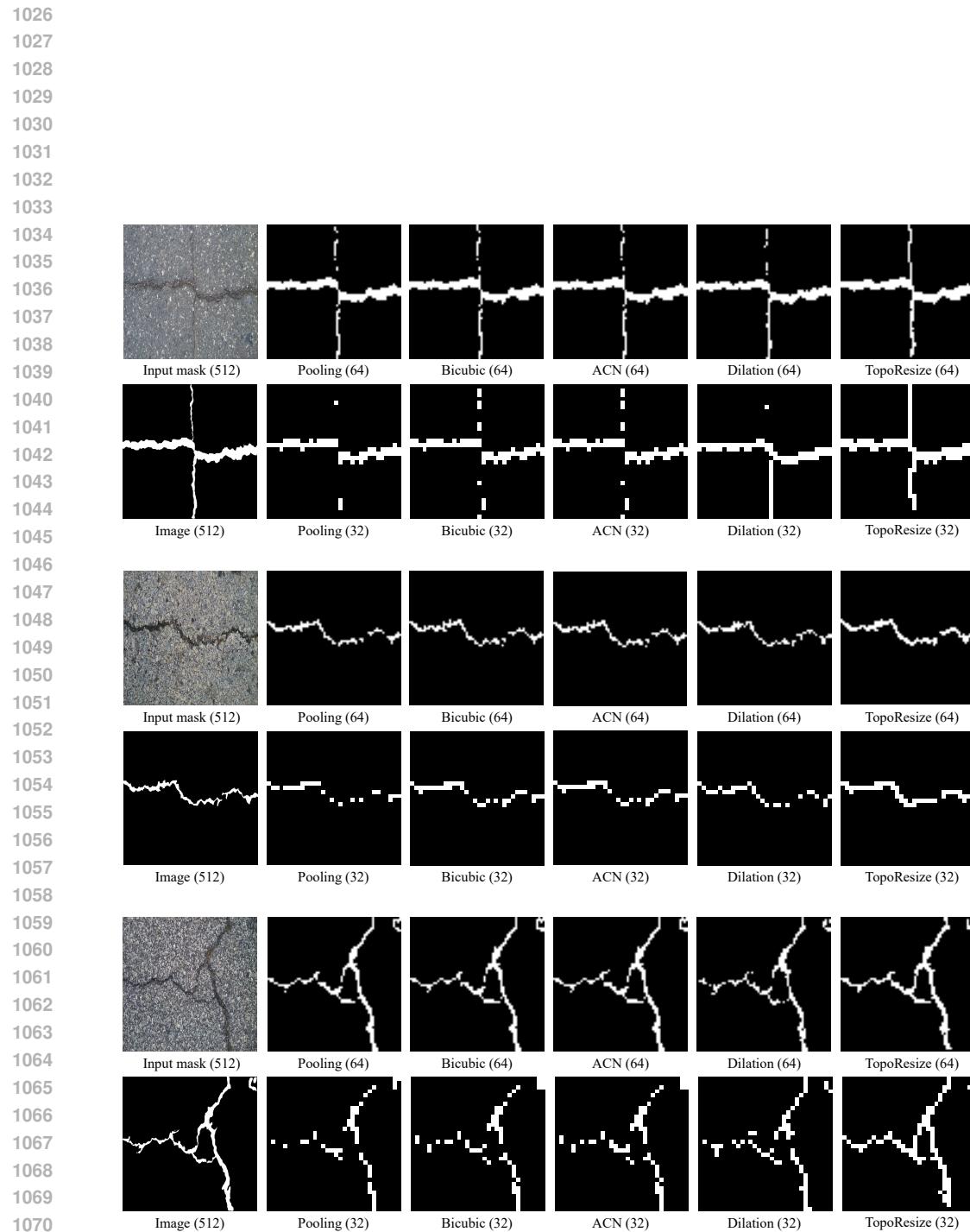


Figure 8: TopoDownsample versus conventional down-sampling methods on two crack masks.

1080 A.6 ADDITIONAL DETAILS FOR EXPERIMENTS
10811082 A.6.1 EXPERIMENTAL SETUP
10831084 **Hyper-parameter settings**1085 All experiments were run on a workstation equipped with two NVIDIA RTX A6000 GPUs (48 GB
1086 each), an AMD EPYC 7513 CPU, and 256 GB RAM. Gurobi 11.0 is used for all MIP optimisations.
10871088 Unless stated otherwise, all hyper-parameters below are shared across *CrackInst1K*, *DeepCrack* and
1089 *CRACK500*.
1090

- **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 $k \sim \mathcal{U}\{0, \dots, 4\}$; maximum crack-pixel ratio $m = 0.8$; step length 2 px; turning angle $\pm 30^\circ$; step budget per start point `min_steps` = 30, `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.

1102 **Synthetic-data generation**
1103

- **SDM (Skeleton→Mask).** UNet depth 4, base channels 128; cosine β schedule, $T=1000$ 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.

1118 **Downstream instance segmentation**
1119

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 \times 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 $3\times$ 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.6.2 MORE EXPERIMENTAL RESULTS

More detectors performance on CrackInstSynth

Table 4 reports the $\text{mAP}_{50}^{\text{seg}}$ 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.

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. 9 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.

1188

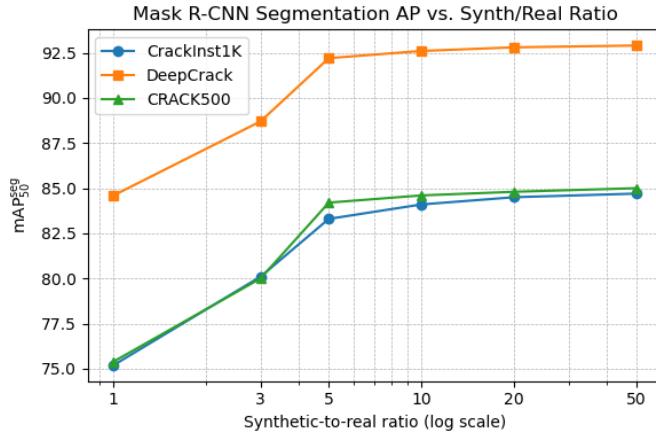
1189 Table 5: Cross-dataset generalisation of Mask R-CNN between datasets. $\text{mAP}_{50}^{\text{seg}}$.

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

1198

1199

1200



1214

1215 Figure 9: Mask R-CNN segmentation AP versus the amount of synthetic data, plotted on a log-
1216 scaled *x*-axis.

1217

1218

A.6.3 MORE VISUALIZATIONS

1219

1220

1221

1222

1223

1224

1225

Fig. 10 shows skeleton→mask→image examples with four material domains. Columns (a)–(b) illustrate that physics-driven growth and the Skeleton2Mask diffusion recover realistic widths and branching. Columns (c)–(f) demonstrate prompt-driven style transfer: *tunnel*, *pavement*, *brick*, and *concrete*. Across rows, crack pixels remain aligned to the same mask and connectivity is unchanged, indicating *label-faithful, multi-style* synthesis that supports our cross-dataset generalization experiments.

1226

1227

A.6.4 PSG QUANTITATIVE ALIGNMENT: FULL STATISTICS

1228

1229

1230

1231

To demonstrate that PSG-generated masks match real masks in key *curvilinear* and *topological* statistics beyond visuals, we report full distribution distances (Wasserstein W_1 and Kolmogorov–Smirnov) and robust-location summaries (medians/IQRs).

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

Feature	W_1	KS stat	KS <i>p</i> -value	n_{real}	n_{psg}
length_mean	13.3595	0.0300	0.688	1134	1134
tortuosity_mean	0.0143	0.4109	4.58×10^{-86}	1134	1134
turning_std	0.00512	0.0591	0.0382	1134	1134
curvature_std	0.00830	0.3298	5.51×10^{-55}	1134	1134
deg3p_share	0.00633	0.0926	1.19×10^{-4}	1134	1134
β_0	0.0203	0.00706	1.000	1134	1134
β_1	0.0176	0.0106	1.000	1134	1134

Table 6: REAL vs. PSG: Wasserstein W_1 and KS statistics across curvilinear/topology features.

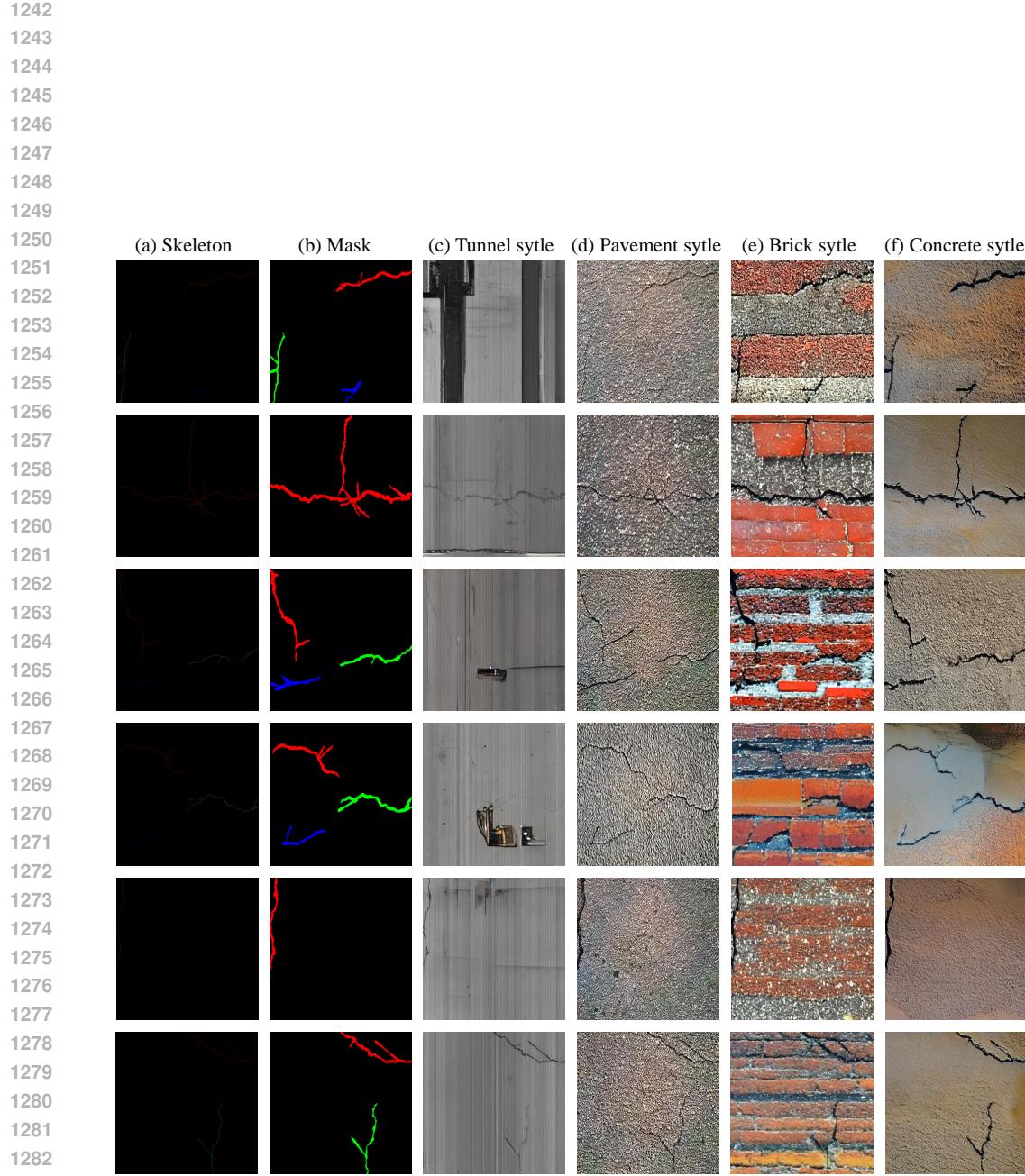


Figure 10: **Additional visualizations across materials (CrackInstSynth).** Columns: (a) *Skeleton* after physics-driven growth; (b) *Mask* produced by the *Skeleton*→*Mask* diffusion; (c) *Tunnel* style; (d) *Pavement* style; (e) *Brick* style; (f) *Concrete* style. The same mask (instance IDs in colour) is rendered into different materials by switching text prompts. TC-ControlNet transfers background appearance while preserving per-instance topology and width.

In Tab. 6, W_1 distances are small across all features, and KS statistics are low for most metrics; $\beta_{0,1}$ match particularly well (KS $p=1.0$). For tortuosity and curvature, KS p -values are tiny due to the large sample size, but the corresponding W_1 magnitudes remain very small, indicating negligible practical shift.

Feature	REAL			PSG		
	median	p25	p75	median	p25	p75
length_mean	208.79	74.48	558.36	207.51	80.04	557.89
tortuosity_mean	1.000	1.000	1.000	1.000	1.000	1.027
turning_std	0.5391	0.5173	0.5567	0.5364	0.5140	0.5600
curvature_std	0.3441	0.3409	0.3453	0.3425	0.3328	0.3530

Table 7: REAL vs. PSG: medians and IQRs (p25–p75). Small absolute W_1 and aligned robust summaries indicate close distributional agreement.

In Tab. 7, medians and IQRs of key curvilinear statistics are closely aligned between REAL and PSG; deviations are minor and directionally consistent with the small W_1 in Tab. 6.

Tables 6 and 7 jointly show that PSG preserves the distribution of curvilinear (length, tortuosity, turning/curvature) and topological ($\beta_{0,1}$) descriptors: effect sizes (W_1) are small and robust summaries (medians/IQRs) align closely with REAL data. While some KS tests become significant under large n , the practical shifts are negligible. These results substantiate the claim that PSG is *physics-consistent* beyond visual plausibility and does not distort the data-generating geometry.

A.6.5 PSG SENSITIVITY AND RESOLUTION-AWARE SCALING

To clarify what matters to tune in PSG and how to port settings across resolutions, we sweep key parameters at 1024^2 and report a simple resolution rule with normalized geometry metrics.¹

Parameter sweep (@ 1024^2 , $m=0.10$). Defaults: $k=4$, $\ell=2$ px, max=200 iters (max=100 · ℓ), $\theta \in \{\pm 15^\circ, \pm 30^\circ, \pm 45^\circ\}$.

Param (values)	Best	Aggregate W range	Δ vs. default	# within +5%	Note
k (2,4,8)	4	13.3904 → 13.7421	+0.0% → +2.6%	3/3	Highly insensitive
max (100,200,300)	200	13.3904 → 13.4400	+0.0% → +0.4%	3/3	Runtime knob; stable
ℓ px (1,2,3)	2	13.3904 → 14.8500	+0.0% → +10.9%	2/3	Affects lengths
θ ($\pm 15^\circ, \pm 30^\circ, \pm 45^\circ$)	$\pm 30^\circ$	13.3904 → 14.5800	+0.0% → +8.9%	2/3	Affects curvature

Table 8: PSG sensitivity at 1024^2 : most parameters are robust; ℓ and θ modulate geometry as expected.

In Tab. 8, varying k or max has negligible effect ($\leq 2.6\%$ and $\leq 0.4\%$ on the aggregate W), whereas ℓ and θ control geometry (length/curvature) with still moderate shifts; most settings remain within $\pm 5\%$ of the default.

Resolution scaling rule. Let $s = \frac{\text{target}}{1024}$; keep (k, m) fixed; scale $\ell = 2s$ px and max = $100 \cdot \ell$.

Resolution (rule-scaled)	Normalized Aggregate W	Δ vs. 1024^2	Note
512^2 ($s=0.5, \ell=1$, max=100)	0.0449	+1.9%	Normalized by size
1024^2 ($s=1.0, \ell=2$, max=200)	0.04399	0.0%	Baseline
2048^2 ($s=2.0, \ell=4$, max=400)	0.0434	-1.3%	Normalized by size

Table 9: Resolution-aware scaling keeps geometry statistics stable after size normalization.

In Tab. 9), after size normalization, the aggregate geometry drift remains within $\pm 2\%$ from 512^2 to 2048^2 , indicating that the linear rule preserves the distributional shape across resolutions.

¹Aggregate metric: sum of 1D Wasserstein distances over length_mean, tortuosity_mean, turning_std, curvature_std, deg3p_share; lower is better.

1350 For practical tuning, prioritize ℓ and θ ; the defaults $k=4$ and $\text{max}=200$ are sufficient. The simple
1351 scaling rule ($\ell=2s$, $\text{max}=100\ell$) maintains stable geometry statistics across resolutions while keeping
1352 the parameterization minimal.
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403