sensors

MDPI

Article

Prompt-Free Metal Crack Video Segmentation Using an Adapted SAM2

Wenqin Jiang [1], Boyue Xu [1], Ruonan Zhang [2,3], Pengyu Wei [2,3], Ke Wang [4] and Jia Bei [1,*]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China; jiangwq@smail.nju.edu.cn (W.J.); xuby@smail.nju.edu.cn (B.X.)
[2] China Ship Scientific Research Center, Wuxi 214082, China; zhangrn@cssrc.com.cn (R.Z.); wei_pengyu@163.com (P.W.)
[3] National Key Laboratory of Ship Structural Safety, Wuxi 214082, China
[4] School of Naval Architecture and Ocean Engineering, Jiangsu University of Science and Technology, Zhenjiang 212100, China; ecsibelle@126.com
* Correspondence: beijia@nju.edu.cn

Abstract

Given sequential metal-surface images, crack video segmentation seeks accurate crack masks and stable crack-tip localization across frames. Thin cracks with low contrasts are difficult to distinguish from the background, and specimen motion further complicates consistent segmentation across frames. To address these challenges, we propose a prompt-free Segment Anything Model 2 (SAM2) framework integrating spatial crack segmentation with a temporal consistency-assisted state update (TCSU) mechanism. The proposed method uses sparse high-level low-rank adaptation and encoder stage-end residual adapters to adapt SAM2 to thin crack structures, with boundary consistency loss strengthening boundary-sensitive training. TCSU verifies spatial evidence against registered appearance change and the previous state and recursively maintains the sequence state, including the crack mask and tip position. On SteelCrack, the proposed method achieved 78.41% miIoU and 87.02% miDice. Online experiments demonstrate the effectiveness of the proposed method in monitoring crack initiation and propagation during fatigue testing.

Keywords: metal crack video segmentation; prompt-free SAM2; parameter-efficient adaptation; temporal verification; crack-tip localization

1. Introduction

Steel structures are widely used in bridges, ships, industrial facilities, and large-scale civil infrastructure [1]. Under cyclic loading, corrosion, environmental exposure, and local stress concentration, surface cracks may initiate and extend, and their observable morphology provides important evidence for condition assessment and maintenance decision-making [2]. Steel cracks are typically thin and sparse, with low contrasts and indistinct boundaries. These features make them difficult to distinguish from surface textures and reflections, hindering accurate crack identification and delineation. During fatigue loading, specimen displacement and changes in crack opening further complicate the consistent localization and monitoring of crack initiation and propagation. To address these challenges, we aim to monitor crack initiation and propagation by estimating crack masks, states, and tip locations from successive images, as illustrated in Figure 1.

Check for updates

Academic Editor: Jongmyon Kim

Received: 8 October 2026
Revised: 3 November 2026
Accepted: 5 November 2026
Published: 10 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

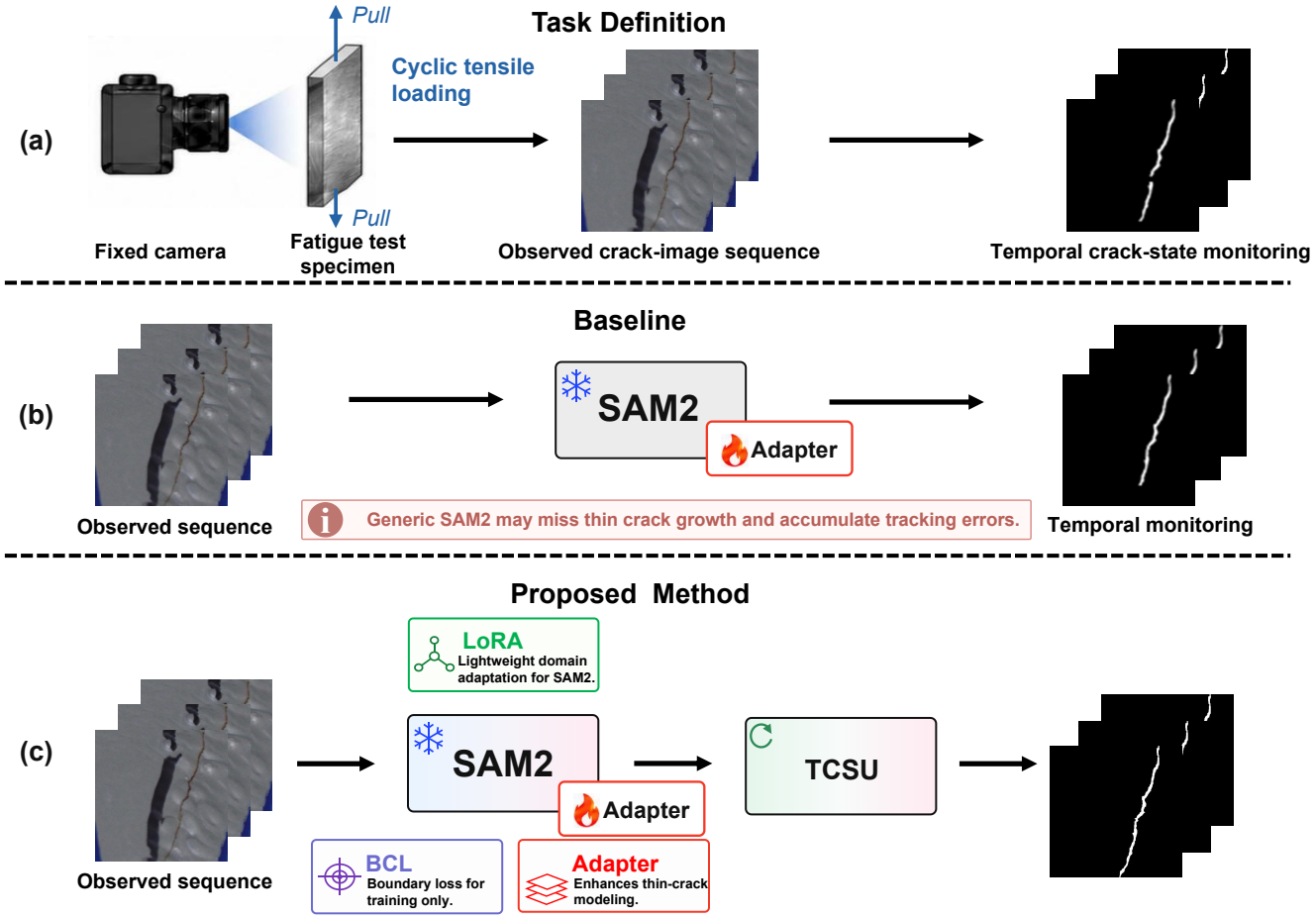


Figure 1. Task definition and comparison of segmentation methods. (**a**) Task definition: Monitoring crack initiation and propagation in a fatigue-test specimen during cyclic tensile loading. (**b**) Baseline: Independent frame-by-frame segmentation using SAM2 with adapters. (**c**) Proposed method: Combining crack-specific spatial adaptation with temporal verification.

Manual visual inspection is labor-intensive and sensitive to operator experience, while conventional image-processing methods depend strongly on acquisition conditions and manually selected parameters. Computer vision and deep learning therefore provide a promising route for extracting crack regions from acquired images and supporting repeatable observation [3].

Deep-learning-based crack segmentation has progressed from convolutional encoder–decoder models [4] to Transformer and hybrid convolutional–Transformer architectures [5,6], with multi-scale fusion, global dependency modeling, and boundary refinement used to improve thin-crack recovery. Segmentation foundation models, represented by the Segment Anything Model (SAM) [7] and its successor Segment Anything Model 2 (SAM2) [8], offer strong generic visual priors and have motivated downstream adaptation studies [9,10]. Adapting these priors to metal cracks requires separating weak, elongated structures from competing surface texture while preserving narrow boundaries.

Most crack-segmentation studies evaluate independent images. When an image model is applied frame by frame to a video sequence, small appearance disturbances may cause intermittent masks, isolated false positives, or implausible tip jumps. Reliable crack video segmentation therefore requires both accurate spatial evidence and sequence-level acceptance based on observed history, so that crack state and tip position remain temporally stable. A video sequence of a fatigue-loaded specimen provides the history needed to check

whether weak crack evidence persists and whether new segments agree with the previously observed path.

Motivated by these observations, we formulate metal crack video segmentation from an observed sequence. At each time step, the proposed framework estimates a crack mask, crack state, and crack-tip location. The proposed method operates within a local observation domain and integrates sparse high-level low-rank adaptation (LoRA), encoder stage-end residual adapters, and boundary consistency loss (BCL) as a unified spatial-evidence stage. Temporal consistency-assisted state update (TCSU) subsequently evaluates the candidate evidence using registered appearance change, crack-path connectivity, directional and tip continuity, and recent observation history. The crack tip then provides recursive feedback for updating the observation domain in the next frame.

We evaluate frame-level perception on the public SteelCrack benchmark and illustrate sequence-level behavior on private open-world videos. The proposed method achieves miIoU and miDice scores of 78.41% and 87.02%, respectively. Qualitative comparisons with frame-wise SAM2 show that the proposed method is more effective at segmenting cracks as they propagate over time.

The main contributions of this paper are summarized as follows.

- The proposed method combines sparse high-level LoRA, stage-end residual adapters, and training-only boundary-consistency supervision for thin-crack delineation.
- We design TCSU to verify spatial candidates using registered appearance change, crack-path continuity, tip consistency, and recent observation history, producing recursively updated masks, states, tips, and observation domains.
- We evaluate frame-level segmentation on SteelCrack with component and LoRA-placement ablations and qualitatively compare the complete monitoring framework with frame-wise SAM2 on two private specimen sequences.

2. Related Work

2.1. Metal Crack Segmentation in Image Sequences

Dense segmentation has progressed from convolutional encoder–decoder models such as fully convolutional networks, U-Net, and DeepLab [4,11,12] to hierarchical Transformer and mask-classification frameworks such as SegFormer and Mask2Former [13,14]. Metal cracks remain difficult targets because they are thin, sparse, low-contrast, and frequently embedded in textured surfaces. Dong et al. [1] introduced an encoder–decoder network for pixel-level fatigue-crack segmentation in large steel-structure images, while Li et al. [15] investigated fine-crack extraction from full-scale steel-bridge images with complicated backgrounds. These studies established the effectiveness of deep segmentation for structural inspection but also exposed the difficulty of preserving weak and discontinuous crack regions. Subsequent work has emphasized boundary quality, multi-scale representation, and long-range structural continuity. He et al. [3] incorporated boundary guidance into steel-structure crack segmentation, and Jin et al. [16] used conditional modulation and hierarchical dual-path learning to combine contextual features. CrackMamba [17] and the topology-aware Mamba model of Zuo et al. [18] use state-space modeling to capture spatial dependencies in crack images. By aggregating contextual information along feature sequences, these methods support the representation of elongated crack structures and their connectivity.

Temporal crack characterization has also been studied using digital image correlation and related deformation-measurement techniques. Réthoré et al. [19] estimated crack-tip positions and stress-intensity factors from displacement fields, Paysan et al. [20] combined robotic imaging with digital image correlation for crack-growth measurement, and

Vasco-Olmo et al. [21] quantified crack-tip opening displacement during loading. These approaches provide mechanically interpretable measurements when stable texture, calibration, and displacement estimation are available. By contrast, applying an image segmentation model independently to a sequence can produce intermittent detections, fragmented paths, and unstable crack-tip locations.

The above studies motivate a unified treatment of spatial crack extraction and temporal verification. In this work, frame-level crack evidence is evaluated together with historical observations to maintain a stable crack representation across frames. The formulation connects image segmentation with recursive sequence-level updating while retaining an image-only evaluation protocol for public crack benchmarks.

2.2. SAM2-Based Crack Segmentation and Adaptation

SAM [7] introduced prompt-conditioned segmentation based on large-scale visual pre-training, and SAM2 [8] extended this paradigm to images and videos through streaming memory and cross-frame propagation. Their generic visual priors are valuable for transferring segmentation knowledge, yet cracks remain challenging because of their small foreground ratio, narrow width, irregular boundaries, and long-range connectivity.

Several studies have adapted SAM-family models to crack and structural-damage segmentation. Ge et al. [22] combined SAM with Adapter and LoRA fine-tuning in CrackSAM to improve crack segmentation across infrastructure scenes and materials. Ye et al. [23] introduced the MCrack1300 instance-segmentation dataset and investigated automatically executable SAM-based models for masonry damage and cracks. Li et al. [24] proposed TopoSAM, which supplemented SAM with topology-aware representation and background-invariant learning for cross-domain crack segmentation. These studies demonstrate the value of foundation-model priors while showing that thin, topologically sensitive targets still require crack-specific adaptation.

Recent multimodal studies further demonstrate that task-specific guidance can extend SAM-family priors beyond generic RGB segmentation. Li et al. [25] proposed KAN-SAM, which introduces thermal features into SAM2 through Kolmogorov–Arnold Network adapters and uses mutually exclusive random masking to improve RGB–thermal salient-object detection. Hou et al. [9] developed HyPSAM, combining dynamic cross-modal fusion with hybrid prompts to refine object completeness and boundaries in RGB–thermal scenes.

SAM-based adaptation also supports specialized domains and reduced annotation requirements. Medical SAM Adapter [26] introduces space–depth transposition and prompt-conditioned adapters for medical segmentation. SSFam [27] combines a frozen SAM encoder with modality-aware modulation and a siamese decoder to connect scribble-prompted training with prompt-free testing. For RGB–depth salient-object detection, DiffRGBD [28] combines an adapted SAM2 encoder with conditional diffusion, using the denoising time step to modulate the conditioning features. Si et al. [29] use SAM to generate dense pseudo annotations from scribbles and state-space interaction to construct cross-modal conditions for diffusion-based refinement. These approaches adapt representation, supervision, or iterative prediction to their target tasks. Diffusion time steps describe the denoising process and are distinct from successive camera observations of crack growth.

Research directly based on SAM2 for crack segmentation remains comparatively limited. SC-SAM2 [30] adapts SAM2 to fatigue-crack segmentation on steel bridges, while SAM2-UNet [10] demonstrates more generally that the SAM2 encoder can support task-oriented dense prediction. Existing applications focus primarily on spatial segmentation accuracy, leaving the temporal integration of frame-wise SAM2 evidence for stable video segmentation underexplored.

Parameter-efficient fine-tuning offers a practical means of adapting foundation models without updating all pre-trained parameters. LoRA and AdaLoRA learn low-rank attention updates [31,32], visual prompt tuning and AdaptFormer introduce compact trainable representations or residual modules [33,34], and SAM-Adapter transfers lightweight adaptation to SAM [35].

These studies establish both lightweight SAM adaptation and prompt-free inference in related tasks. Our framework targets prompt-free metal crack video segmentation from a single visual stream. It combines crack-specific parameter-efficient adaptation and boundary supervision with sequence-level evidence verification, thereby linking frame-wise crack extraction to recursively updated video masks.

2.3. LoRA-Based Adaptation for Crack Segmentation

LoRA enables the parameter-efficient adaptation of pre-trained segmentation models by learning low-rank weight updates while keeping the corresponding pre-trained weights frozen. This allows segmentation features to be adapted to specific domains with a limited number of trainable parameters. For crack segmentation, LoRA provides a means of adapting foundation-model representations to thin structures, weak boundaries, and complex backgrounds.

Recent studies have explored a LoRA-based adaptation of SAM-family models for crack and structural-damage segmentation. Ge et al. [22] developed CrackSAM by adapting SAM through adapter-based and LoRA-based fine-tuning and evaluating their combination. Their study examined the effects of adaptation capacity on segmentation accuracy and cross-dataset generalization, demonstrating robustness under noise and challenging imaging conditions. Guo et al. [36] combined LoRA-based feature adaptation with a dedicated crack segmentation head to enable automatic segmentation and evaluated the method across eight crack datasets. Ye et al. [23] used LoRA to fine-tune the SAM encoder for masonry damage segmentation. They developed two automatic segmentation approaches, one connecting the adapted encoder to alternative decoders and the other introducing a learnable self-generating prompter, and evaluated brick and crack instances using MCrack1300. More recently, Hendrichovsky et al. [37] investigated LoRA-assisted SAM2 adaptation in CRAX for interactive annotation of cracks and other thin structures. They compared decoder-only training, joint LoRA and decoder training, and full encoder–decoder fine-tuning, examining segmentation quality, cross-domain generalization, and annotation effort.

However, these studies mainly focus on image segmentation, with limited attention to monitoring crack development over time. In the proposed method, LoRA adapts SAM2 to the segmentation requirements of steel-crack monitoring, helping identify thin cracks against complex surface backgrounds and maintain consistent segmentation during crack initiation and propagation.

3. Methods

3.1. Unified Crack Video-Segmentation Framework

The framework shown in Figure 2 couples spatial crack evidence with recursive temporal verification. Within the current observation domain, the proposed method produces a candidate probability map using sparse high-level LoRA and encoder stage-end residual adapters. BCL supervises only training. TCSU then verifies the candidate against registered appearance change and the previous state, updates the mask, crack state, and tip, and uses the crack tip to define the next observation domain. Successive frames retain sufficient spatial overlap for crack-path comparison, and local registration compensates for inter-frame specimen displacement.

The framework represents sequence history through the previous state and updates it recursively. On independent-image benchmarks, the proposed method is evaluated with TCSU disabled.

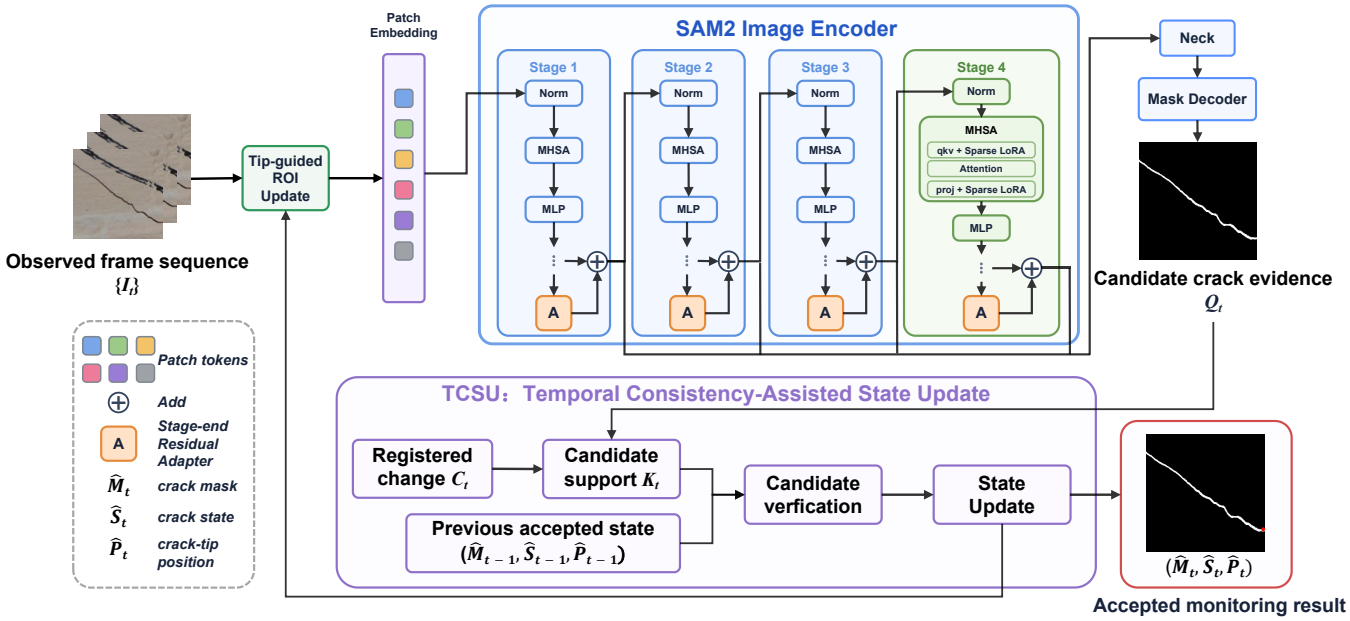


Figure 2. The unified crack video-segmentation framework. Patch embedding converts the cropped image into tokens. The SAM2 encoder extracts hierarchical image features. LoRA adapts the encoder to the crack-segmentation task. Stage-end residual adapters refine local crack geometry and structural continuity. The neck fuses multi-scale features for the mask decoder. The mask decoder produces candidate crack evidence. TCSU verifies this evidence using registered appearance change and sequence history, then updates the crack mask, state, and tip position. The updated tip and path guide the observation region for the next frame. The red dot marks the estimated crack-tip position.

For each video frame, the framework estimates a crack mask; a crack state selected from Normal, Candidate, and Crack; and a crack-tip position. The segmentation network of the proposed method is trained on individual images with ground-truth crack masks and derived boundary labels. It predicts a crack-probability map for the current observation region. TCSU checks this evidence against registered appearance change and the previous crack path before updating the mask, state, and tip.

3.2. Spatial Segmentation in the Proposed Method

The proposed method uses a prompt-free SAM2 pathway, sparse high-level LoRA, and encoder stage-end residual adapters. Given the current crop, the encoder and neck produce hierarchical features, and the mask decoder predicts a single low-resolution crack-logit map. The logits are resized to the input resolution and passed through a sigmoid function. The resulting probabilities are mapped to full-image coordinates to form candidate evidence for TCSU. TCSU verifies the candidate evidence and updates the monitoring state. BCL provides boundary supervision during training.

3.2.1. Prompt-Free SAM2 Baseline

We configure SAM2 [8] for prompt-free binary segmentation by replacing point, box, and mask prompts with a fixed, empty, sparse embedding and a learnable dense prior. We retain the Hiera image-encoder trunk, feature-pyramid neck, and mask decoder of SAM2.

For an RGB image, hierarchical encoder features pass through the neck and mask decoder to produce one dense crack-mask prediction. High-resolution features remain avail-

able to the decoder for thin-structure localization. The neck supplies the image embedding and two higher-resolution feature maps. The decoder runs with multimask output disabled.

The proposed method performs prompt-free crack segmentation. TCSU uses registered appearance change and the previous state to verify current crack evidence and update the crack mask, crack state, and tip position.

For baseline training, the main encoder trunk and prompt encoder remain frozen, while the neck, full mask decoder, and learnable no-mask embedding are optimized. This configuration preserves the generic SAM2 representation and adapts the decoding pathway to steel cracks. Sparse high-level LoRA, encoder stage-end residual adapters, and boundary-consistency supervision are then incorporated into this baseline.

3.2.2. Sparse High-Level LoRA

The prompt-free SAM2 baseline provides generic visual priors, while steel cracks require domain-specific sensitivity to thin, sparse structures on complex metallic surfaces. We therefore apply sparse high-level LoRA to a small set of selected encoder blocks. The low-rank branches adapt high-level semantic representations with limited trainable parameters and preserve the remaining pre-trained encoder weights.

For a frozen pre-trained weight matrix $\mathbf{W}$, LoRA introduces a trainable low-rank update:

$$\mathbf{y} = \left(\mathbf{W} + \frac{\alpha}{r}\mathbf{BA}\right)\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors, $\mathbf{A}$ and $\mathbf{B}$ are trainable low-rank factors, r is the rank, and α controls the update scale.

LoRA is inserted into the first three blocks of Stage 4 in the SAM2 encoder. Within these blocks, low-rank branches are applied to the joint query–key–value and attention output projections. The original projection weights remain frozen while the low-rank factors are trained. The up-projection is initialized to zero, so the adapted projections initially preserve the pre-trained mapping.

3.2.3. Multi-Branch Crack-Aware Feature Refinement

A residual adapter is placed after the last block of each encoder stage to refine local crack geometry and structural continuity. It operates on the stage-output features to enhance the representation of thin, elongated cracks and weak boundaries. The residual connection incorporates this refinement into the existing encoder features. These adapters provide spatial refinement across the encoder hierarchy, complementing the high-level semantic adaptation performed by LoRA in selected late encoder blocks.

As shown in Figure 3, the adapter applies layer normalization and GELU, reduces the channel dimension, and processes the features through parallel depthwise convolution branches for local, directional, and wider contextual modeling. Appropriate padding preserves the spatial dimensions across branches, allowing their outputs to be summed element-wise. The fused features are projected back to the original channel dimension and added to the input through a learnable scaled residual connection.

3.2.4. Boundary Consistency Loss

Steel cracks are thin, sparse, and highly boundary-sensitive. Region-level supervision can tolerate small spatial deviations that interrupt a narrow crack path or remove fine branches. We use BCL as a training-only auxiliary objective that constrains the boundary derived from the predicted crack mask. The inference network retains a single-mask output. Figure 4 summarizes the complete dual-supervision training flow.

The crack-mask branch compares the predicted logits with the ground-truth mask using the sum of Dice and IoU losses. The sigmoid function converts the logits into a crack-probability map, from which a differentiable morphological gradient extracts the predicted soft boundary. The ground-truth boundary is dilated to form a tolerant target, reducing sensitivity to small spatial deviations.

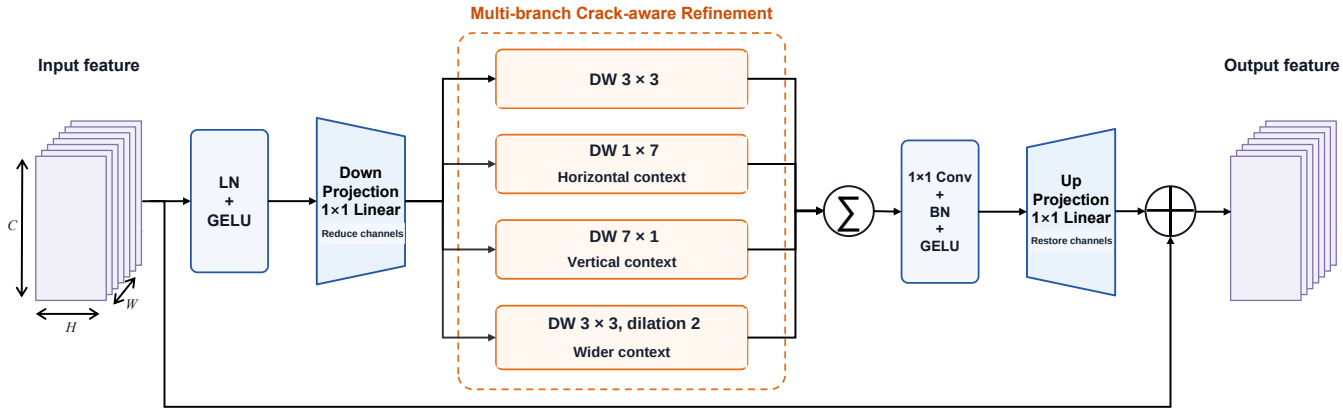


Figure 3. The architecture of the encoder stage-end residual adapter. The dashed Multi-branch Crack-aware Refinement region is the internal spatial-refinement unit of the adapter. After normalization and activation, channel-reduced features are processed by complementary depthwise convolution branches for local, directional, and contextual modeling. The fused response is projected back to the input channel dimension and added to the identity shortcut.

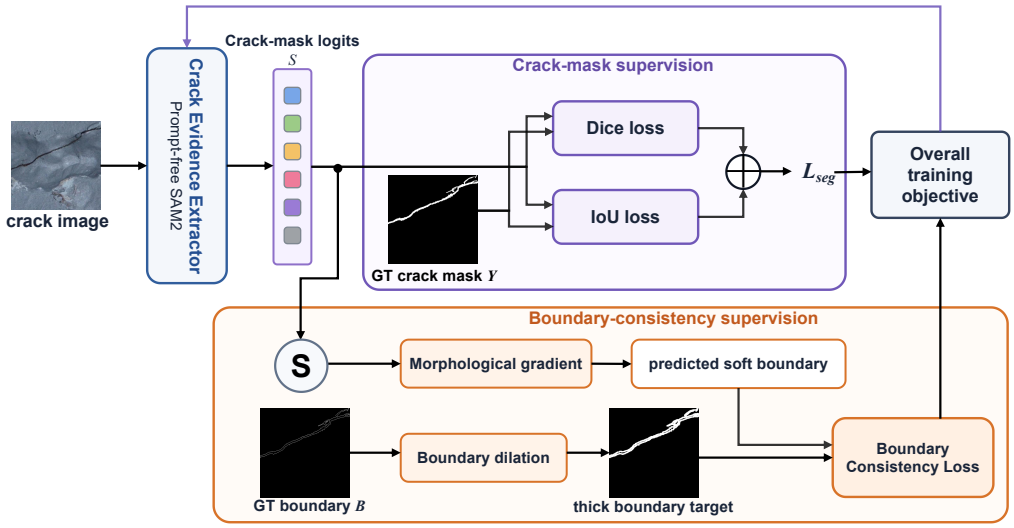


Figure 4. The dual-supervision training workflow for the proposed method. The **upper** branch supervises crack-mask predictions with Dice and IoU losses. The **lower** branch supervises predicted soft boundaries with dilated boundary targets. Boundary supervision is used only during training. The colored squares schematically represent crack-mask logits.

Let $\hat{b}_i$ and $\tilde{b}_i$ denote the predicted soft-boundary value and the dilated target value at pixel i, respectively. Their Dice loss defines boundary consistency:

$$\mathcal{L}_{\mathrm{BCL}} = 1 - \frac{2\sum_i \hat{b}_i \tilde{b}_i + \epsilon}{\sum_i \hat{b}_i + \sum_i \tilde{b}_i + \epsilon}, \tag{2}$$

where i indexes pixels and ϵ is a smoothing constant that keeps the Dice ratio defined for empty boundaries.

BCL is activated after a warm-up period so that the network first learns the main crack region and subsequently receives boundary-sensitive refinement. After this warm-up, the overall training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \lambda_{\text{b}} \mathcal{L}_{\text{BCL}}, \tag{3}$$

where $\mathcal{L}_{\text{seg}}$ is the sum of Dice and IoU losses and λ_{b} weights the boundary term. BCL is used only during training.

3.3. Temporal Consistency-Assisted State Update

TCSU converts frame-level crack evidence into a sequence state through recursive observation-domain localization, spatiotemporal evidence verification, and crack-state transition modeling. The following subsections describe these three operations.

3.3.1. Recursive Observation-Domain Localization

For the first observation, sequence initialization supplies the observation domain and the initial state. All subsequent domains are updated recursively from the latest state. Subsequent regions are centered at the last crack tip and oriented along the recent crack path. Positive forward, backward, and transverse margins define the crop extent. The crop is passed to the segmentation network of the proposed method, and its probability map is mapped back to full-image coordinates. An unverified observation retains the previous mask and tip.

3.3.2. Spatiotemporal Evidence Verification

Temporal verification is formulated as a consistency test between the spatial candidate and the observed sequence history. Since camera or specimen motion can create false temporal differences, the current observation domain is first registered to a stable reference region around the notch or specimen texture, producing $\bar{I}_t$. A baseline image I_t^{b} is maintained from frames classified as Normal. The normalized change response is

$$C_t(\mathbf{x}) = \frac{|\bar{I}_t(\mathbf{x}) - I_t^{\text{b}}(\mathbf{x})|}{\sigma_t(\mathbf{x}) + \epsilon}, \tag{4}$$

where R_t is the current observation region, $\mathbf{x} \in R_t$, $C_t(\mathbf{x})$ is the normalized change response, $\bar{I}_t(\mathbf{x})$ is the registered current intensity, $I_t^{\text{b}}(\mathbf{x})$ is the Normal-state baseline intensity, $|\cdot|$ denotes absolute difference, and $\sigma_t(\mathbf{x})$ is the local intensity scale. The positive constant ϵ prevents division by zero and has the same intensity units as σ_t. The baseline is updated only in the Normal state by

$$I_{t+1}^{\text{b}} = (1 - \eta) I_t^{\text{b}} + \eta \bar{I}_t, \tag{5}$$

where I_{t+1}^{b} is the updated baseline and $\eta \in [0, 1]$ is its update rate. The baseline is frozen in Candidate and Crack states to avoid absorbing a growing crack into the background model.

Let $Q_t = \sigma(\mathcal{F}(I_t|_{R_t}; \Theta))$ be the SAM crack probability mapped to full-image coordinates. Candidate construction combines the current spatial evidence with the registered change response inside the local tip neighborhood, using the previous mask $\hat{M}_{t-1}$ and tip $\hat{\mathbf{p}}_{t-1}$ as context to form a candidate set K_t. For $K_t \neq \emptyset$, the provisional tip is the point with the largest longitudinal coordinate,

$$\mathbf{p}_t^* = \arg\max_{\mathbf{x} \in K_t} u_t(\mathbf{x}), \tag{6}$$

where $u_t(\mathbf{x}) = (\mathbf{x} - \hat{\mathbf{p}}_{t-1})^\top \mathbf{d}_t$ and $\mathbf{d}_t$ is the recent propagation direction. Candidate acceptance jointly requires path connectivity, tip localization, and displacement consistent with the recent propagation direction. An empty candidate set fails verification and retains the previous mask and tip.

3.3.3. Crack-State Transition Modeling

Let $z_t \in \{0, 1\}$ denote the verification outcome. The three-state update introduces temporal hysteresis: Normal changes to Candidate when $z_t = 1$. Candidate changes to Crack when at least m of the most recent k observations, including the current one, are verified. Otherwise, k_{reset} consecutive failures reset it to Normal. Other cases retain the previous state. The voting parameters satisfy $1 \le m \le k$. The consecutive-failure counter ℓ_t is carried between observations and initialized with $\ell_0 = 0$. Short unsupported gaps after confirmation retain the last mask and tip.

The crack tip is updated only from verified evidence. To suppress pixel-level jitter, it is temporally smoothed as

$$\hat{\mathbf{p}}_t = \beta \mathbf{p}_t^* + (1 - \beta)\hat{\mathbf{p}}_{t-1}, \tag{7}$$

where $\beta \in [0, 1]$ is the smoothing coefficient, $\mathbf{p}_t^*$ is the provisional tip, and $\hat{\mathbf{p}}_{t-1}$ is the previous crack tip. The update is subject to the longitudinal monotonicity constraint, so the crack tip does not move backward along $\mathbf{d}_t$.

Algorithm 1 summarizes TCSU. Verified support is accumulated with the crack path in registered coordinates. Otherwise, the last confirmed mask and tip are retained.

Algorithm 1 Temporal Consistency-Assisted State Update (TCSU).

Require: Frame I_t, baseline I_t^{b}, previous tuple $(\hat{M}_{t-1}, \hat{s}_{t-1}, \hat{\mathbf{p}}_{t-1})$
Require: Recent evidence history $\mathcal{H}_{t-1}$, previous failure count ℓ_{t-1}, segmentation network of the proposed method $\mathcal{F}$
Require: Voting-window length k, confirmation count m, and reset count k_{reset}
Ensure: Updated tuple $(\hat{M}_t, \hat{s}_t, \hat{\mathbf{p}}_t)$, baseline I_{t+1}^{b}, history $\mathcal{H}_t$, count ℓ_t, next domain R_{t+1}
Localize the observation using the crack tip and path.
$\mathbf{d}_t \leftarrow \textsc{EstimateDirection}(\hat{M}_{t-1}, \mathcal{H}_{t-1})$
$R_t \leftarrow \textsc{UpdateDomain}(\hat{\mathbf{p}}_{t-1}, \mathbf{d}_t)$
$\bar{I}_t \leftarrow \textsc{Register}(I_t|_{R_t})$
$Q_t \leftarrow \textsc{MapToImage}(\sigma(\mathcal{F}(I_t|_{R_t})))$
Combine spatial evidence with registered appearance change.
$C_t \leftarrow \textsc{AppearanceChange}(\bar{I}_t, I_t^{\text{b}})$
$K_t \leftarrow \textsc{ConstructCandidate}(Q_t, C_t, R_t, \hat{M}_{t-1}, \hat{\mathbf{p}}_{t-1})$
$z_t \leftarrow 0$
if $K_t \neq \emptyset$ **then**
  $\mathbf{p}_t^* \leftarrow \arg\max_{\mathbf{x} \in K_t} (\mathbf{x} - \hat{\mathbf{p}}_{t-1})^\top \mathbf{d}_t$
  # Check connectivity, tip localization, and directional displacement.
  $z_t \leftarrow \textsc{VerifyCandidate}(K_t, \mathbf{p}_t^*, \hat{M}_{t-1}, \hat{\mathbf{p}}_{t-1}, \mathbf{d}_t)$
end if
$W_t \leftarrow \text{Tail}_k(\textsc{Outcomes}(\mathcal{H}_{t-1}) \,\|\, [z_t])$
$\ell_t \leftarrow 0$ **if** $z_t = 1$ **else** $\ell_{t-1} + 1$
$\hat{s}_t \leftarrow \hat{s}_{t-1}$
# Confirm persistent evidence and reset unsupported candidates.
if $\hat{s}_{t-1} =$ Normal **and** $z_t = 1$ **then**
  $\hat{s}_t \leftarrow$ Candidate
else if $\hat{s}_{t-1} =$ Candidate **and** $\textsc{sum}(W_t) \ge m$ **then**
  $\hat{s}_t \leftarrow$ Crack
else if $\hat{s}_{t-1} =$ Candidate **and** $\ell_t \ge k_{\text{reset}}$ **then**
  $\hat{s}_t \leftarrow$ Normal
end if
$(\hat{M}_t, \hat{\mathbf{p}}_t) \leftarrow (\hat{M}_{t-1}, \hat{\mathbf{p}}_{t-1})$
if $z_t = 1$ **then**
  # Accumulate verified support in registered coordinates.
  $\hat{M}_t \leftarrow \hat{M}_{t-1} \cup K_t$
  $\hat{\mathbf{p}}_t \leftarrow \textsc{StabilizeTip}(\mathbf{p}_t^*, \hat{\mathbf{p}}_{t-1}, \mathbf{d}_t)$
  # Smooth the tip while preventing backward longitudinal motion.
end if
$I_{t+1}^{\text{b}} \leftarrow I_t^{\text{b}}$
if $\hat{s}_t =$ Normal **then**
  # Update the background only in the Normal state.
  $I_{t+1}^{\text{b}} \leftarrow \textsc{UpdateBaseline}(I_t^{\text{b}}, \bar{I}_t)$
end if
$\mathcal{H}_t \leftarrow \textsc{Append}(\mathcal{H}_{t-1}, z_t, \hat{\mathbf{p}}_t)$
$R_{t+1} \leftarrow \textsc{UpdateDomain}(\hat{\mathbf{p}}_t, \mathbf{d}_t)$
return $(\hat{M}_t, \hat{s}_t, \hat{\mathbf{p}}_t), I_{t+1}^{\text{b}}, \mathcal{H}_t, \ell_t, R_{t+1}$

Segmentation masks can additionally support pixel-domain summaries such as skeleton path length and apparent mask width. These optional descriptors characterize the observed sequence.

4. Experiments

4.1. Benchmark and Evaluation Metrics

The main quantitative experiments use SteelCrack [3], a public pixel-level benchmark collected from steel-structure projects under diverse acquisition conditions. The dataset comprises 3300 training, 525 validation, and 530 test images, each measuring 512×512 pixels.

SteelCrack provides independently annotated images without temporal correspondence or crack-evolution labels. Accordingly, we evaluate the proposed method with TCSU disabled on this image benchmark.

We report miIoU and miDice, computed over N test images as follows:

$$\text{miIoU} = \frac{1}{N}\sum_{n=1}^{N}\frac{|\mathcal{P}_n \cap \mathcal{G}_n| + \epsilon}{|\mathcal{P}_n \cup \mathcal{G}_n| + \epsilon}, \tag{8}$$

$$\text{miDice} = \frac{1}{N}\sum_{n=1}^{N}\frac{2\sum_{\mathbf{x}} p_n(\mathbf{x}) g_n(\mathbf{x}) + \epsilon}{\sum_{\mathbf{x}} p_n(\mathbf{x}) + \sum_{\mathbf{x}} g_n(\mathbf{x}) + \epsilon}, \tag{9}$$

where $\mathcal{P}_n$ and $\mathcal{G}_n$ are the predicted and ground-truth foreground pixel sets for test image n, $p_n(\mathbf{x})$ is the normalized grayscale prediction, $g_n(\mathbf{x})$ is the binary target at pixel $\mathbf{x}$, and N is the number of test images. The notation $|\cdot|$ denotes set cardinality, and $\cap$ and $\cup$ denote intersection and union. The smoothing constant is $\epsilon = 10^{-6}$.

For SAM2 and the proposed method, sigmoid predictions are normalized separately for each image using min–max scaling. IoU is computed from masks binarized at a threshold of 0.5, and Dice is computed directly from the normalized grayscale predictions.

Model complexity is measured by parameter counts and MACs. M and G denote millions and billions, respectively.

4.2. Implementation Details

The proposed method is implemented in Python 3.10.0 with PyTorch 2.6.0 using SAM2.1 Hiera-L and custom LoRA and convolutional adapter modules. Experiments use one NVIDIA GeForce RTX 5090 GPU, with SteelCrack images resized to 512 × 512. Training uses AdamW with a batch size of 6 and an initial learning rate of 6×10^{-5}. The neck, mask decoder, no-mask embedding, and stage-end residual adapters are trainable. Sparse LoRA is applied to the first three blocks of Stage 4 with rank 8, scaling parameter $\alpha = 16$, and dropout 0.05. BCL has a weight of 0.1 and is activated after an initial warm-up period.

Given the limited availability of public datasets capturing the temporal evolution of metal cracks, segmentation accuracy is evaluated on the public SteelCrack dataset. Videos recorded during online tests are then used to qualitatively assess crack-path continuity and crack-tip localization over time. The frame-wise SAM2 baseline and the complete framework process the same local image crops.

4.3. Quantitative Comparison

Compared methods include convolutional, Transformer, Mamba-based state-space, and boundary-guided architectures, together with SAM-based crack segmenters on SteelCrack [3].

Table 1 compares segmentation accuracy and model complexity on the SteelCrack benchmark.

Table 1. Quantitative comparison with state-of-the-art methods on the SteelCrack benchmark. The dataset comprises 3,300 training, 525 validation, and 530 test images, each measuring 512 × 512 pixels.

Method	Year	miIoU (%)	miDice (%)	Parameters (M)	Trainable Parameters (M)	MACs (G)
U-Net [4]	2015	69.81	76.85	31.04	31.04	219.01
Attention U-Net [38]	2018	71.25	77.54	34.88	34.88	266.54
DeepLabv3+ [12]	2018	69.04	69.45	59.34	59.34	88.84
U-Net++ [39]	2020	72.23	78.37	9.16	9.16	138.63
CE-Net [40]	2019	76.00	81.54	29.00	29.00	35.60
SCRN [41]	2019	73.23	78.91	25.23	25.23	31.92
TransUNet [6]	2021	64.34	72.55	67.87	67.87	129.96
CrackSeU-B [42]	2023	70.42	80.50	3.19	3.19	11.22
CrackSeU-L [42]	2023	71.66	81.24	4.62	4.62	28.22
DconnNet [5]	2023	74.73	83.40	28.38	28.38	24.79
CrackFormer-II [43]	2023	66.11	80.32	4.90	4.90	82.70
SAM-Adapter [35]	2023	67.33	78.30	86.49	–	88.73
BGCrack [3]	2024	<u>77.16</u>	85.33	2.32	2.32	15.76
TAMAMBA [18]	2024	75.25	85.87	75.00	75.00	147.00
ASPS [44]	2024	60.92	72.64	37.04	37.04	44.61
SAM2 [8]	2024	75.03	84.35	216.92	4.77	202.86
CMHA [16]	2025	–	<u>86.12</u>	14.96	14.96	73.65
SAM2-UNet [10]	2026	75.65	85.04	216.40	5.91	–
Ours	–	**78.41**	**87.02**	217.57	5.42	203.75

Bold values indicate the highest score for each segmentation metric among all listed methods. Underlined values indicate the highest score for each metric among the compared prior methods.

The proposed method achieves miIoU and miDice scores of 78.41% and 87.02%, respectively, the highest listed values for both metrics. It outperforms U-Net and CE-Net on both metrics and also exceeds the boundary-guided BGCrack and Mamba-based TAMAMBA. Its miIoU is higher than BGCrack's 77.16%, while its miDice surpasses CMHA's 86.12%, the highest reported miDice among the compared prior methods.

The proposed method also achieves higher a miIoU and miDice than SAM-Adapter and SAM2-UNet. Compared with the prompt-free SAM2 baseline, it raises miIoU from 75.03% to 78.41% and miDice from 84.35% to 87.02%. These results support the effectiveness of combining spatial adaptation and boundary supervision for steel-crack segmentation.

Complexity Analysis

The proposed method contains 217.57 M parameters, of which 5.42 M are trainable, and requires 203.75 G MACs for a 512 × 512 input. Most parameters belong to the frozen SAM2 prior. The trainable subset comprises the prompt-free decoding pathway, sparse LoRA branches, and stage-end adapters. BCL is used only to construct the training objective and therefore introduces neither inference-time parameters nor additional inference modules. Trainable parameters account for 2.49% of the total. Compared with SAM2, the trainable parameter count increases from 4.77 M to 5.42 M. The computational cost increases from 202.86 G to 203.75 G MACs, a relative increase of 0.44%.

4.4. Ablation Study

The cumulative ablation starts from the prompt-free SAM2 baseline and successively introduces the stage-end residual adapters, sparse high-level LoRA, and BCL.

4.4.1. Component Contributions

Table 2 shows that the prompt-free SAM2 baseline, which fine-tunes the neck and mask decoder and achieves 75.03% miIoU and 84.35% miDice. Introducing the four stage-end residual adapters increases the scores to 76.96% and 85.86%, respectively. Sparse LoRA in the high-level attention blocks further raises them to 77.45% miIoU and 86.33% miDice. With boundary-consistency supervision, the proposed method reaches 78.41% miIoU and 87.02% miDice. Each cumulative addition improves the reported scores of the preceding configuration.

Table 2. Cumulative component ablation on the SteelCrack test set. The best results are highlighted in bold.

Adapter	Sparse LoRA	BCL	miIoU (%)	miDice (%)	Parameters (M)	Trainable Parameters (M)
			75.03	84.35	216.92	4.77
✓			76.96	85.86	217.43	5.26
✓	✓		77.45	86.33	217.57	5.42
✓	✓	✓	**78.41**	**87.02**	217.57	5.42

4.4.2. Effect of LoRA Insertion Strategy

Table 3 compares sparse high-level, individual-stage, and all-stage LoRA placements at rank 8, with stage-end adapters and BCL retained. Sparse placement adapts the first three blocks of Stage 4, whereas the full Stage 4 variant adapts all four blocks. To assess the parameter cost of different LoRA insertion strategies, we report both LoRA-only and total trainable parameter counts.

Table 3. LoRA insertion strategy comparison.

Placement	Blocks	LoRA Parameters (M)	Trainable Parameters (M)	miIoU (%)	miDice (%)
Sparse high-level	3	0.16	5.42	78.41	87.02
Stage 1	2	0.01	5.27	77.71	86.39
Stage 2	6	0.08	5.34	77.94	86.62
Stage 3	36	0.99	6.25	78.24	86.92
Stage 4	4	0.22	5.48	77.36	86.16
All stages	48	1.31	6.56	78.59	87.21

All-stage adaptation achieves 78.59% miIoU and 87.21% miDice, compared with 78.41% and 87.02% for the sparse reference. It uses 1.31 M LoRA parameters, compared with 0.16 M for the sparse design, while total trainable parameters increase from 5.42 M to 6.56 M. The sparse design uses fewer trainable parameters, while all-stage adaptation gives higher segmentation scores.

4.4.3. LoRA Rank Sensitivity

Table 4 examines LoRA rank in the first three encoder blocks of Stage 4. We use $r \in \{4, 8, 16\}$ and $\alpha = 2r$, keeping the scaling ratio $\alpha/r = 2$, dropout 0.05, stage-end adapters, and BCL weight 0.1 unchanged. We additionally evaluate LoRA ranks 4 and 16.

Increasing the rank from 4 to 16 quadruples LoRA parameters from 80,640 to 322,560 and raises the total trainable parameters from 5.34 M to 5.58 M, while miIoU increases from 77.53% to 77.72% and miDice from 86.30% to 86.50%. The rank-8 configuration achieves miIoU and miDice scores of 78.41% and 87.02%, respectively, at an intermediate parameter budget.

Table 4. LoRA rank comparison.

Rank r	α	LoRA Parameters (M)	Trainable Parameters (M)	miIoU (%)	miDice (%)
4	8	0.08	5.34	77.53	86.30
8 (Ours)	16	0.16	5.42	78.41	87.02
16	32	0.32	5.58	77.72	86.50

4.5. Backbone Representation Visualization

To examine where the parameter-efficient adaptation changes the spatial encoding of SAM2, we compare aligned Hiera features from the prompt-free SAM2 baseline and the proposed method. At each spatial location, adaptation-induced change is summarized by the channel-wise Euclidean distance between the two feature tensors. Figure 5 visualizes this distance for Stages 2–4 after bilinear upsampling to the input resolution. Each map is independently normalized between its 2nd and 98th percentiles for display. Consequently, color indicates the relative spatial distribution within one panel rather than an absolute magnitude across stages. The ground-truth contour is shown only as a localization reference and is not involved in the feature-distance calculation.

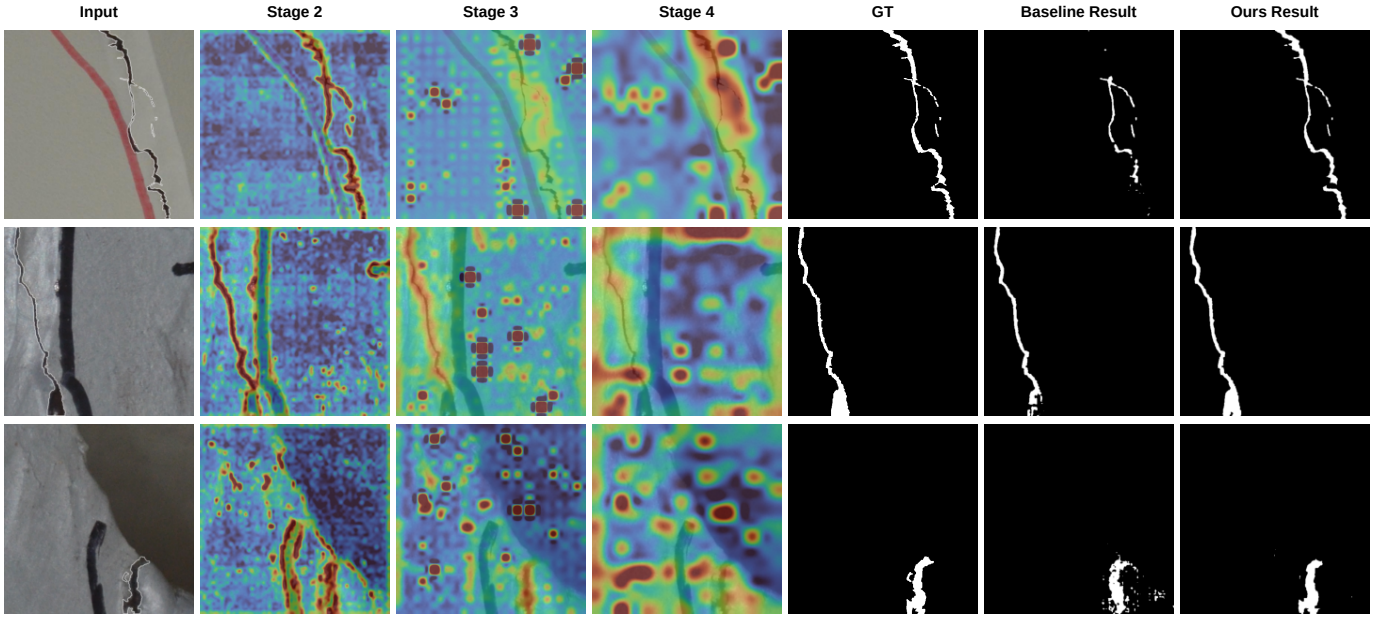


Figure 5. The feature visualization on representative SteelCrack test images. From left to right: Input image with the ground-truth contour, Hiera Stage 2, Stage 3, and Stage 4 feature-distance maps, GT, baseline result, and our result. Colors in the feature-distance maps indicate the relative magnitude of feature differences between the baseline and the proposed method.

In these representative cases, the Stage 2 changes follow relatively fine crack neighborhoods, while the deeper stages provide coarser responses because of their progressively reduced spatial resolution. The changes remain associated with crack-bearing regions across the hierarchy, and the complete-model masks recover more continuous crack structures than the baseline predictions in the same examples. The visualization provides spatial evidence that the learned adaptation modifies crack-relevant representations.

4.6. Qualitative Comparison

Figure 6 compares the proposed method with the fine-tuned prompt-free SAM2 baseline. In these examples, the complete model preserves longer connected paths and finer

boundaries for small or low-contrast cracks, whereas the baseline more often omits narrow segments and local branches.

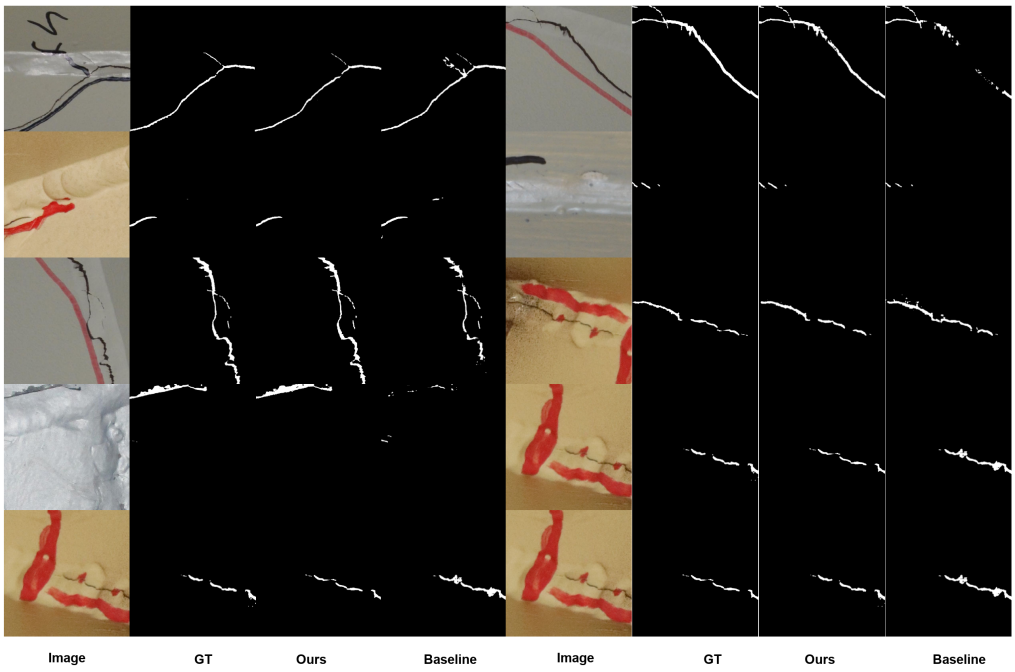


Figure 6. The qualitative comparison of crack segmentation results on SteelCrack. In each group, the columns denote the input image, ground truth, our result, and the fine-tuned prompt-free SAM2 baseline, respectively. In the ground-truth and predicted masks, white denotes crack regions and black denotes the background.

4.7. Open-World Sequential Evaluation

The open-world evaluation examines a welded-joint surface specimen and a side-surface steel specimen, whose appearances differ substantially from the public benchmark. The welded-joint sequence contains a low-contrast surface crack, while the side-surface sequence contains a notch-adjacent region with dense surface texture. Both spatial variants receive the same source observation and local crop. TCSU additionally uses a fixed surrounding reference band for registration and change verification throughout each sequence.

Fatigue tests were conducted on an Instron 8802 testing machine at 5–10 Hz. A fixed MV-CH650-90YM-M58S-NN camera manufactured by Hangzhou Hikrobot Co., Ltd., Hangzhou, China, was positioned 15 cm from the specimen, using a 50 mm lens and auxiliary illumination. The camera has a resolution of 9344 × 7000 pixels. Videos were acquired in real time at 70 fps to record crack initiation and propagation. The observed cracks typically ranged from 5 to 20 mm in length, with an image scale of approximately eight pixels per 0.1 mm.

Figure 7 presents representative observations from both sequences. White pixels in the binary masks indicate crack regions identified by the methods. Frame-wise SAM2 responds to the welded-joint crack in clear views but shows boundary and width variation between observations. The side-surface specimen provides weaker spatial evidence because of the notch-adjacent textured surface. TCSU integrates verified sequence history with the current evidence, yielding a more coherent crack path in the illustrated observations.

Figure 8 complements the side-surface images with the temporal evolution of predicted crack length, measured in pixels along a simplified polyline from the crack root to the tip. Circles in Figure 8 mark selected nodes representing changes in predicted crack length, and straight lines connect consecutive nodes to show the trend. The initial unplotted interval corresponds to fatigue loading before the crack appears in the video. The plotted length

increases from approximately 85 to 117 pixels in the early part of the recorded curve, changes more gradually around 120–130 pixels through much of the middle interval, and rises to approximately 210 pixels near the end.

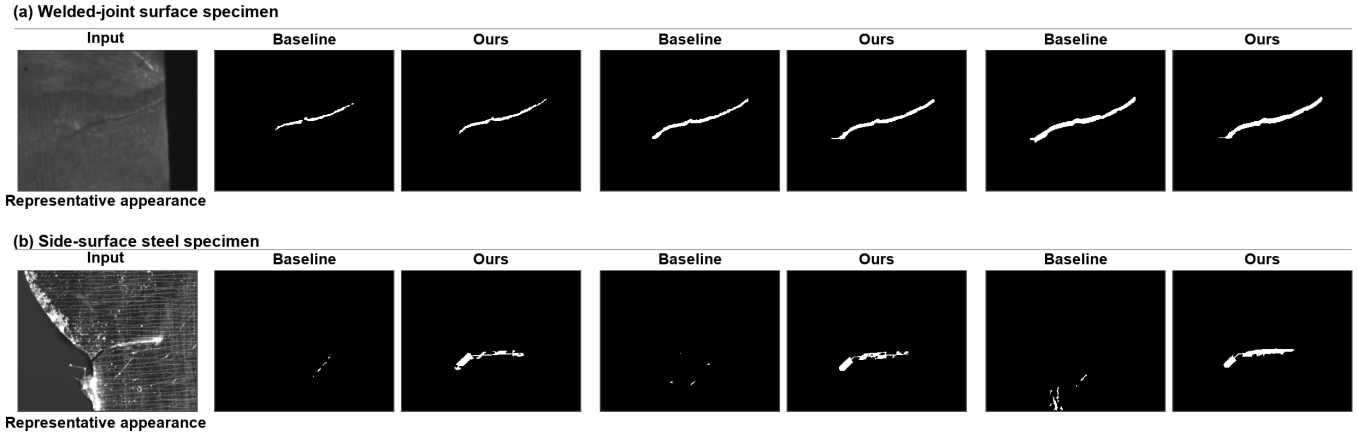


Figure 7. The open-world qualitative comparison on a welded-joint surface specimen and a side-surface steel specimen. The first panel in each row shows a representative input frame. The remaining panels compare frame-wise SAM2 with the complete framework using binary crack masks.

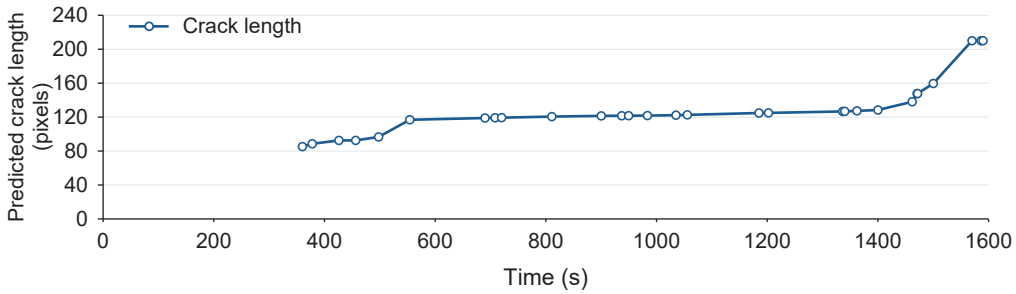


Figure 8. The predicted crack length over time for the side-surface specimen. The displayed nodes summarize the length changes rather than every sampled video observation. Length is expressed in pixels.

5. Discussion

The SteelCrack results show that a small trainable subset can adapt the frozen SAM2 prior to thin-crack segmentation. The proposed method attains a miIoU of 78.41% and a miDice score of 87.02% with 5.42 M trainable parameters. The cumulative ablation shows progressively higher scores after adding stage-end adapters, sparse high-level LoRA, and BCL. Feature visualizations and qualitative masks support these findings by showing adaptation near crack-bearing regions and improved recovery of narrow branches. The 203.75 G MAC cost nevertheless remains close to the baseline because the frozen backbone is still executed during inference.

The sequential examples evaluate temporal behavior beyond the independent-image benchmark. TCSU accepts spatial evidence only when it agrees with registered appearance change and the accumulated crack state, producing more coherent paths on the welded-joint and side-surface sequences. Across the two surface conditions, the examples show qualitatively more coherent crack paths. However, sequence-level accuracy cannot be quantified without dense temporal masks and independently annotated tip trajectories. SteelCrack therefore supports the spatial evaluation, and the video tests provide qualitative evidence for temporal consistency.

Limitations and Future Work

Figure 9 shows two representative failure cases of the proposed method: over-segmentation and incomplete detection. In the over-segmentation case, the predicted crack is wider than the ground-truth annotation. In the incomplete-detection case, the prediction misses part of a thin annotated crack segment.

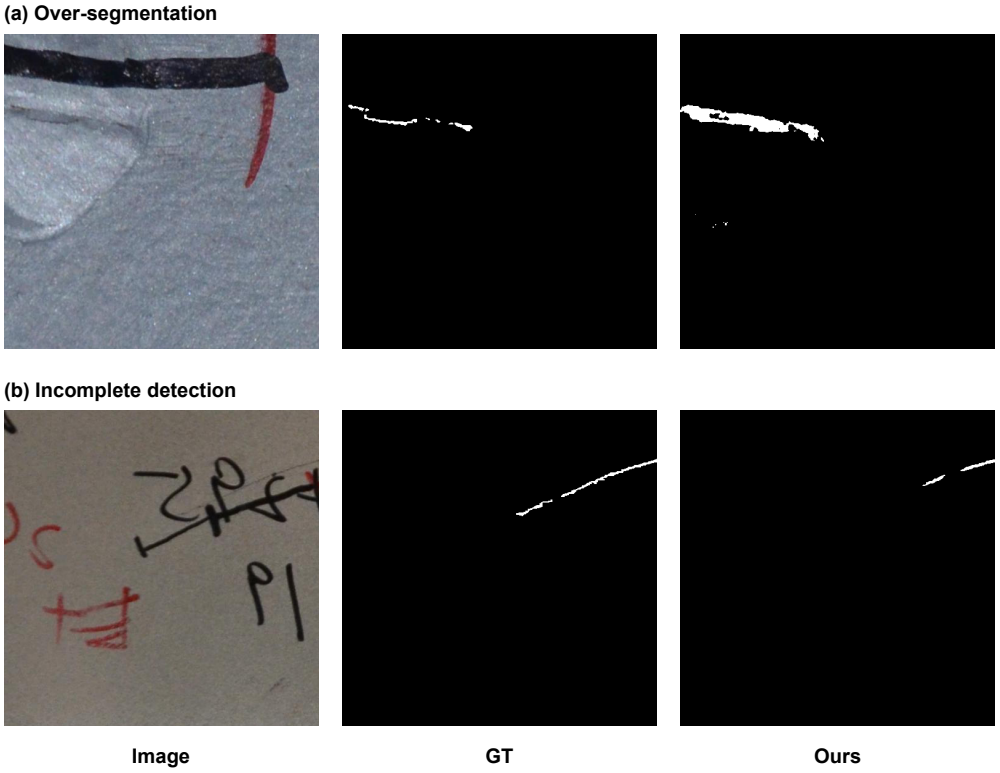


Figure 9. The representative spatial-segmentation failure cases on SteelCrack. The columns labeled Image, GT, and Ours show the input image, ground-truth mask, and prediction of the proposed method, respectively. Panel (**a**) shows an excessive predicted crack width. Panel (**b**) shows the incomplete recovery of a thin annotated segment.

Future work should obtain dense temporal masks and independent tip annotations, evaluate registration and illumination robustness and end-to-end latency, and validate calibrated crack-extension measurements.

6. Conclusions

This paper presented a prompt-free SAM2 framework for metal crack video segmentation that combined crack-specific spatial adaptation with temporal state updating. The proposed method integrates sparse high-level LoRA, stage-end residual adapters, and boundary-consistency supervision to recover thin and discontinuous crack structures with limited trainable parameters. TCSU verifies the current spatial evidence against registered appearance change and the sequence state to maintain coherent crack paths and tip estimates across observations. On SteelCrack, the proposed method achieves a miIoU of 78.41% and a miDice score of 87.02% with 5.42 M trainable parameters, while the qualitative metal crack video tests show more coherent crack paths than frame-wise inference.

Author Contributions: Conceptualization, W.J., B.X. and J.B.; methodology, W.J. and B.X.; software, W.J. and B.X.; validation, W.J., B.X., R.Z. and K.W.; formal analysis, W.J. and B.X.; investigation, R.Z., P.W. and K.W.; resources, R.Z. and P.W.; data curation, W.J., B.X., R.Z. and P.W.; visualization, W.J. and B.X.; writing—original draft preparation, W.J. and B.X.; writing—review and editing, R.Z., P.W., K.W. and J.B.; supervision, J.B.; project administration, P.W. and J.B.; funding acquisition, P.W. and J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project of Virtual-Real Fusion Analysis Technology for Ship Structure Fatigue Performance (WDZC70202030206).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Access to the private sequential videos and associated experimental records is restricted under project confidentiality. The public SteelCrack benchmark can be obtained from its original source. The custom code developed for this study is not publicly available.

Acknowledgments: The authors thank the project team for providing the specimen materials and experimental validation records used in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dong, C.; Li, L.; Yan, J.; Zhang, Z.; Pan, H.; Catbas, F.N. Pixel-Level Fatigue Crack Segmentation in Large-Scale Images of Steel Structures Using an Encoder–Decoder Network. *Sensors* **2021**, *21*, 4135. [CrossRef] [PubMed]
2. Deng, J.; Singh, A.; Zhou, Y.; Lu, Y.; Lee, V.C.S. Review on Computer Vision-Based Crack Detection and Quantification Methodologies for Civil Structures. *Constr. Build. Mater.* **2022**, *356*, 129238. [CrossRef]
3. He, Z.; Chen, W.; Zhang, J.; Wang, Y.H. Crack Segmentation on Steel Structures Using Boundary Guidance Model. *Autom. Constr.* **2024**, *162*, 105354. [CrossRef]
4. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
5. Yang, Z.; Farsiu, S. Directional Connectivity-Based Segmentation of Medical Images. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 11525–11535.
6. Chen, J.; Mei, J.; Li, X.; Lu, Y.; Yu, Q.; Wei, Q.; Luo, X.; Xie, Y.; Adeli, E.; Wang, Y.; et al. TransUNet: Rethinking the U-Net Architecture Design for Medical Image Segmentation through the Lens of Transformers. *Med. Image Anal.* **2024**, *97*, 103280. [CrossRef] [PubMed]
7. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 4015–4026.
8. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. In Proceedings of the 13th International Conference on Learning Representations, Singapore, 24–28 April 2025.
9. Hou, R.; Li, X.; Ren, T.; Zhou, D.; Wu, G.; Cao, J. HyPSAM: Hybrid Prompt-Driven Segment Anything Model for RGB-Thermal Salient Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *36*, 2697–2712. [CrossRef]
10. Xiong, X.; Wu, Z.; Tan, S.; Li, W.; Tang, F.; Chen, Y.; Li, S.; Ma, J.; Li, G. SAM2-UNet: Segment Anything 2 Makes Strong Encoder for Natural and Medical Image Segmentation. *Vis. Intell.* **2026**, *4*, 2. [CrossRef]
11. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
12. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder–Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
13. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proceedings of the 35th Conference on Neural Information Processing Systems, Online, 6–14 December 2021; Volume 34, pp. 12077–12090.

14. Cheng, B.; Misra, I.; Schwing, A.G.; Kirillov, A.; Girdhar, R. Masked-Attention Mask Transformer for Universal Image Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 1290–1299.
15. Li, Z.; Zhu, H.; Huang, M. A Deep Learning-Based Fine Crack Segmentation Network on Full-Scale Steel Bridge Images with Complicated Backgrounds. *IEEE Access* **2021**, *9*, 114989–114997. [CrossRef]
16. Jin, Y.; Liu, H.; Cheng, X.; Du, C. Crack Segmentation in Civil Infrastructure Using Conditional Modulation and Hierarchical Dual-Path Networks. *Eng. Struct.* **2025**, *343*, 121162. [CrossRef]
17. Cai, W.; Wang, X.; Xue, Y.; Ma, Y.; Wu, J.; Ge, Z.; Wang, B. CrackMamba with Normalized Soft-Frangi-Filter Enhancement Towards Accurate Crack Segmentation. In Proceedings of the 2025 International Conference on Multimedia Retrieval, Chicago, IL, USA, 30 June–3 July 2025; pp. 43–51.
18. Zuo, X.; Sheng, Y.; Shen, J.; Shan, Y. Topology-Aware Mamba for Crack Segmentation in Structures. *Autom. Constr.* **2024**, *168*, 105845. [CrossRef]
19. Réthoré, J.; Roux, S.; Hild, F. Identification of a Crack Propagation Law by Digital Image Correlation. *Int. J. Fatigue* **2012**, *36*, 146–154. [CrossRef]
20. Paysan, F.; Dietrich, E.; Breitbarth, E. A Robot-Assisted Microscopy System for Digital Image Correlation in Fatigue Crack Growth Testing. *Exp. Mech.* **2023**, *63*, 975–986. [CrossRef]
21. Vasco-Olmo, J.M.; Díaz, F.A.; Antunes, F.V.; James, M.N. Characterisation of Fatigue Crack Growth Using Digital Image Correlation Measurements of Plastic CTOD. *Theor. Appl. Fract. Mech.* **2019**, *101*, 332–341. [CrossRef]
22. Ge, K.; Wang, C.; Guo, Y.; Tang, Y.; Hu, Z.; Chen, H. Fine-Tuning Vision Foundation Model for Crack Segmentation in Civil Infrastructures. *Constr. Build. Mater.* **2024**, *431*, 136573. [CrossRef]
23. Ye, Z.; Lovell, L.; Faramarzi, A.; Ninic, J. SAM-Based Instance Segmentation Models for the Automation of Structural Damage Detection. *Adv. Eng. Inform.* **2024**, *62*, 102826. [CrossRef]
24. Li, X.; Yu, S.; Xiao, S.; Lin, X.; Zhao, Z. A Topology-Aware Segment Anything Model for Domain-Invariant Crack Segmentation. *Eng. Appl. Artif. Intell.* **2026**, *166*, 113688. [CrossRef]
25. Li, X.; Hou, R.; Ren, T.; Wu, G. KAN-SAM: Kolmogorov–Arnold Network Guided Segment Anything Model for RGB-T Salient Object Detection. In Proceedings of the 2025 IEEE International Conference on Multimedia and Expo (ICME), Nantes, France, 30 June–4 July 2025; pp. 1–6.
26. Wu, J.; Wang, Z.; Hong, M.; Ji, W.; Fu, H.; Xu, Y.; Xu, M.; Jin, Y. Medical SAM Adapter: Adapting Segment Anything Model for Medical Image Segmentation. *Med. Image Anal.* **2025**, *102*, 103547. [CrossRef] [PubMed]
27. Liu, Z.; Deng, S.; Wang, X.; Wang, L.; Fang, X.; Tang, B. SSFam: Scribble Supervised Salient Object Detection Family. *IEEE Trans. Multimed.* **2025**, *27*, 1988–2000. [CrossRef]
28. Shi, S.; Li, G.; Cong, R.; Xiao, S.; Lin, W. Diffusion-Driven RGB-D Salient Object Detection with Temporal Modulation. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *36*, 11831–11843. [CrossRef]
29. Si, W.; Li, G.; Shi, S.; Lin, W. Weakly-Supervised RGB-D Salient Object Detection via SAM-Driven Pseudo Annotation and State Space Interaction-Based Diffusion. *IEEE Trans. Multimed.* 2026, early access. [CrossRef]
30. Xu, S.; Li, D.; Shen, R.; Xin, R.; Shi, Y.; Liu, E.; Li, X.; Wang, H. SC-SAM2: Fatigue Crack Segmentation Network of Steel Bridge Based on SAM2. In Proceedings of the 44th Chinese Control Conference (CCC), Chongqing, China, 28–30 July 2025; pp. 1–6.
31. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the 10th International Conference on Learning Representations, Online, 25–29 April 2022.
32. Zhang, Q.; Chen, M.; Bukharin, A.; Karampatziakis, N.; He, P.; Cheng, Y.; Chen, W.; Zhao, T. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In Proceedings of the 11th International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
33. Jia, M.; Tang, L.; Chen, B.C.; Cardie, C.; Belongie, S.; Hariharan, B.; Lim, S.N. Visual Prompt Tuning. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 709–727.
34. Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; Luo, P. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. In Proceedings of the 36th Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 16664–16678.
35. Chen, T.; Zhu, L.; Ding, C.; Cao, R.; Wang, Y.; Zhang, S.; Li, Z.; Sun, L.; Zang, Y.; Mao, P. SAM-Adapter: Adapting Segment Anything in Underperformed Scenes. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Paris, France, 2–6 October 2023; pp. 3367–3375.
36. Guo, Y.; Xu, Y.; Cui, H.; Dang, M.; Li, S. Segment Anything Model-Based Crack Segmentation Using Low-Rank Adaption Fine-Tuning. *Struct. Health Monit.* **2025**, *24*, 2579–2591. [CrossRef]
37. Hendrichovsky, F.; Pickl, R.; Bednar, L.; Soukup, D. CRAX: Parameter-Efficient Fine-Tuning of SAM2 for Interactive Crack Annotation. In Proceedings of the 21st International Conference on Computer Vision Theory and Applications, Marbella, Spain, 9–11 March 2026; Volume 1, pp. 689–700.

38. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention U-Net: Learning Where to Look for the Pancreas. Presented at the 1st International Conference on Medical Imaging with Deep Learning, Amsterdam, The Netherlands, 4–6 July 2018. Available online: https://openreview.net/forum?id=Skft7cijM (accessed on 1 November 2026).
39. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Trans. Med. Imaging* **2020**, *39*, 1856–1867. [CrossRef] [PubMed]
40. Gu, Z.; Cheng, J.; Fu, H.; Zhou, K.; Hao, H.; Zhao, Y.; Zhang, T.; Gao, S.; Liu, J. CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Trans. Med. Imaging* **2019**, *38*, 2281–2292. [CrossRef] [PubMed]
41. Wu, Z.; Su, L.; Huang, Q. Stacked Cross Refinement Network for Edge-Aware Salient Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7264–7273.
42. Chen, W.; He, Z.; Zhang, J. Online Monitoring of Crack Dynamic Development Using Attention-Based Deep Networks. *Autom. Constr.* **2023**, *154*, 105022. [CrossRef]
43. Liu, H.; Yang, J.; Miao, X.; Mertz, C.; Kong, H. CrackFormer Network for Pavement Crack Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9240–9252. [CrossRef]
44. Li, H.; Zhang, D.; Yao, J.; Han, L.; Li, Z.; Han, J. ASPS: Augmented Segment Anything Model for Polyp Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Marrakesh, Morocco, 6–10 October 2024; pp. 118–128.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.