Prism: Harmonizing Missing Modalities via Implicit Structural Alignment on Lightweight Pulse RWKV for Multimodal Crack Segmentation

Hui Liu **Chen Jia** **Fan Shi** **Xu Cheng***
Mianzhao Wang **Jiangpeng Zheng** **Shengyong Chen**
School of Computer Science and Engineering, Tianjin University of Technology
Engineering Research Center of Learning-Based Intelligent System (Ministry of Education)
Key Laboratory of Computer Vision and System (Ministry of Education)
Tianjin, China
liuhui1109@stud.tjut.edu.cn jiachen@email.tjut.edu.cn
shifan@email.tjut.edu.cn xu.cheng@ieee.org
wmz@email.tjut.edu.cn zjp_1997@stud.tjut.edu.cn
sy@ieee.org

Abstract

In multimodal crack segmentation for industrial facilities, the key challenge is preventing missing modalities from degrading pixel-level performance while keeping computation low. Existing methods struggle to harmonize missing modality effects and to efficiently, adaptively perceive cross-modal topology. We propose Prism, resilient to any modality miss, leveraging implicit structural alignment to harmonize missing data for high-quality segmentation at low computational cost. Prism comprises Modal Reconstructor (MoRe), Prompt-as-Knowledge Distillation (PaKD), a Sparse Gated Pulse Propagation Mixer (PulseMixer), and an Uncertainty-guided Connectivity Propagation Fusion module (UCPF). MoRe utilizes the manifolds of available modalities to probabilistically harmonize missing data, empowered by PaKD to master ***where to look*** via prompt distillation. PulseMixer efficiently breaks the static parameterization bottleneck and perceives anisotropic textures via Pulse Propagation Insight (PPI) and sparse gating. UCPF fuses cross-modal cues using uncertainty estimation to produce clear segmentation maps while suppressing background noise. Experiments on three multimodal crack datasets demonstrate strong performance under diverse modality-miss scenarios. On the depth dataset with 90% missing depth modal, Prism achieves 0.8217 in F1 and 0.8478 in mIoU with only 2.62M parameters. The code is available at `https://github.com/Karl1109/Prism`.

1 Introduction

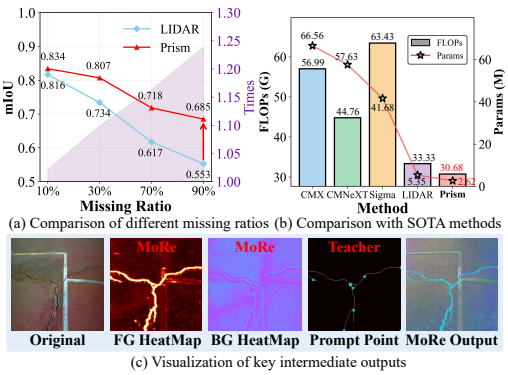
Structural Health Monitoring, especially crack segmentation, is a key safeguard for industrial safety [6, 5], where RGB-based segmentation remains the mainstream automated approach [20, 22, 60]. However, relying solely on RGB is insufficient in complex industrial scenarios, as lighting changes, shadow occlusion, and background texture interference hinder the capture of deep stress distributions and micro-geometric deformations [21]. To address these limitations, multimodal methods exploit cross-modal complementarity by integrating thermophysical cues from infrared thermography, surface textures from polarization, and light-field depth cues to improve segmentation robustness [25].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Most multimodal segmentation relies on the idealized assumption of complete data, contradicting industrial realities where missing modalities are frequent [16, 27, 18]. Moreover, most methods are restricted to simulating entire modality miss, neglecting random intra-modal missingness. Consequently, as shown in Figure 1 (a), existing networks suffer representation collapse and sharp performance drops as the intra-modal missing ratio increases. While recent methods mitigate missing modalities through generative completion [47] or robust multimodal fusion [31], they do not explicitly enforce topological continuity in the latent space, potentially compromising structural perception.

Knowledge Distillation (KD) [46] bridges information gaps by aligning student features with the teacher's structured latent manifold, which embodies physical topological essence to enable reconstruction during missing modalities. However, current paradigms fail to preserve topological integrity. For instance, SKD [37] prioritizes local texture discrimination via pixel-wise contrastive learning, overlooking global geometric coherence. Similarly, CCSD [45] utilizes hierarchical self-distillation to align probability distributions, yet lacks explicit constraints on spatial connectivity or boundary continuity. These approaches teach ***what to predict*** but miss the teacher's saliency-driven focus regarding ***where to look*** for critical topology. Without such guidance, students struggle to reconstruct physically consistent structures.



(a) Comparison of different missing ratios (b) Comparison with SOTA methods

(c) Visualization of key intermediate outputs

Figure 1: The performance of Prism. (a) compares performance across varying missing ratios in RGB. (b) compares the computational requirements. (c) visualizes key intermediate outputs.

Apart from handling missing modalities, balancing efficiency and accuracy on edge devices remains a critical bottleneck. CNN-based methods such as PDCNet [49] and PGDENet [56] offer efficient inference but miss long-range topology due to local inductive biases. Conversely, Transformer variants including TokenFusion [42] and MultiMAE [2] provide global modeling but fail real-time demands due to quadratic complexity. While Mamba architectures, represented by LIDAR [25] and Sigma [40], achieve linear complexity through multi-directional state-space modeling, their scan-based designs remain dependent on explicitly constructed spatial traversal paths; MambaIRv2 [10] enables non-causal modeling via dynamic prompts but incurs additional computational overhead.

The Receptance Weighted Key Value (RWKV) model [33] offers linear complexity via recurrent state transfer. Although Vision RWKV [8] overcomes the limitations of causal modeling, applying it to crack segmentation reveals two theoretical flaws. First, regarding spatial perception, standard RWKV lacks content adaptivity. Strategies such as Q-Shift [8] and Zigzag [4] rely on rigid preset mixing rather than facilitating progressive information propagation along the natural crack topology, causing a misalignment between feature flow and physical structure. Second, at the level of channel interaction, the native operator suffers from a static parameterization bottleneck, lacking fine-grained selectivity, input-dependent sparse gating, and hierarchical intensity control [50]. Consequently, it passively aggregates states, failing to suppress high-frequency noise or amplify critical topological cues, thereby degrading performance in industrial scenarios.

These limitations pose a critical challenge: ***How to harmonize missing modalities for precise segmentation while maintaining low computational overhead?***

To address these challenges, we propose Prism, a lightweight Pulse RWKV network. By leveraging implicit structural space alignment, as shown in Figure 1 (b), Prism achieves superior performance under random modal miss scenarios. Missing modalities are simulated by randomly zeroing out samples within a specific modality at varying ratios, mimicking real-world sensor failures. We design Prompt-as-Knowledge Distillation (PaKD), which capitalizes on SAM2's [35] powerful prompt-driven representation ability exclusively during training as an implicit structural guide to distill point sensitivity into the Modal Reconstructor (MoRe), with Figure 1(c) showing the resulting MoRe outputs. The Sparse Gated Pulse Propagation Mixer (PulseMixer) employs the Pulse Propagation Insight (PPI) strategy for adaptive, skeleton-based propagation. Combined with sparse gating, this breaks the static parameterization bottleneck and maps features to a high-dimensional nonlinear manifold, enhancing topological perception while maintaining linear efficiency. UCPF module utilizes uncertainty estimation to dynamically adjust fusion weights and repair connectivity, ensuring precise segmentation. Our main contributions are as follows:

- We propose Prism, a lightweight pulse RWKV network designed for various missing modalities. Leveraging implicit structural alignment to harmonize missing modalities, it effectively perceives deep topology and stress distributions, generating high-precision segmentation with low computational overhead.
- We design PaKD and MoRe to enhance the harmonization of missing modalities. The PulseMixer, integrating PPI and sparse gating, breaks the static parameterization bottleneck for topological texture extraction, while UCPF leverages uncertainty estimation to fuse cross-modal information for high-quality segmentation.
- Experiments on three datasets under various missing settings demonstrate that Prism outperforms state-of-the-art (SOTA) methods while maintaining minimal computational demands.

2 Related Works

2.1 Crack Segmentation Method

In recent years, unimodal segmentation has dominated crack detection [52, 1]. Crackformer [23] uses scale-wise attention to suppress noise and sharpen boundaries, while SCSegamba [24] employs structure-aware scanning for efficient pixel-level detection. However, these RGB-based methods are limited by the lack of deep-level cues such as stress distribution and spatial depth. Consequently, they struggle with performance degradation under low illumination or stain interference. LIDAR [25] introduced a multimodal approach using mask-guided dynamic scanning. Similarly, in semantic segmentation, MCubeS [19] utilizes region-guided filters to integrate modal radiation characteristics, and CMNeXt [54] achieves superior performance via the efficient Hub2Fuse paradigm.

While these multimodal methods excel with complete data, they rely on the idealized assumption of full modality availability. In industrial reality, sensor failures and hardware constraints frequently cause missing modalities, triggering sharp performance drops. To mitigate this, MCMoE [47] employs a generative MoE mechanism to reconstruct missing inputs, and CHARM [43] achieves collaborative harmonization across modalities through a mutual perception unit. Nevertheless, these missing-modality approaches do not explicitly constrain crack topology, which may compromise consistent structural perception.

2.2 Receptance Weighted Key Value Model

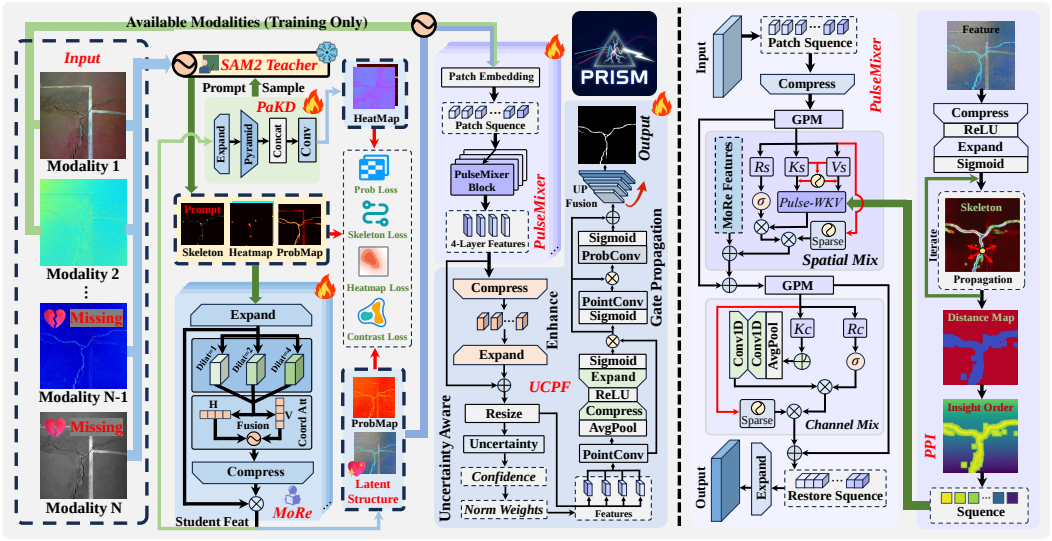
RWKV [33] has emerged as a next-generation architecture, achieving linear complexity via recurrent state transfer. Vision RWKV [8] breaks unidirectional constraints through bidirectional modeling (Bi-WKV) and enhances spatial perception with Q-Shift. RS-RWKV [55] further eliminates fixed scanning bias via random shuffling and a WKV sharing mechanism. To strengthen local dependencies, Dual-LoopLap RWKV [44] introduces Laplace Token Shift, utilizing Laplace decomposition to enhance spatial-frequency interactions. Additionally, GDSR [59] expanded receptive field to capture long-range dependencies through a global-detail reconstruction module.

Despite their progress, existing RWKV models struggle with complex crack topologies. Spatially, they rely on content-agnostic scanning methods such as raster or shuffling that flatten images into 1D sequences, hindering progressive information flow along natural structures. Representatively, the linear attention suffers from a static parameterization bottleneck that projects complex manifold dependencies onto a low rank plane, smoothing out critical details. Without input-aware sparse selectivity and hierarchical re-weighting, the model's blind accumulation fails to prioritize key semantics, allowing weak topological signals to be overwhelmed by historical noise and blurring structural boundaries.

3 Methodology

3.1 Preliminary

As shown in Figure 2, Prism comprises MoRe, PaKD, PulseMixer, and UCPF. Given N multimodal inputs $\mathcal{X} \in \mathbb{R}^{N\times C\times H\times W}$ with missing X_{N-1} and X_N, training uses a frozen SAM2 teacher; under PaKD adaptive sampling, the MoRe student reconstructs latent features to form $\hat{\mathcal{X}}$. For each training sample, SAM2 receives the pixel-wise average of the available three-channel modalities, with missing modalities excluded; if all modalities are absent, it receives zeros. Thus, the teacher and student share identical modality availability. After Patch Embedding, $\hat{\mathcal{X}}$ is tokenized and fed into four PulseMixer Blocks to extract hierarchical topological features, and UCPF then performs progressive uncertainty-guided fusion and upsampling to output $F_{out} \in \mathbb{R}^{1\times H\times W}$.



(a) Network Architecture of Prism (b) Detail of PulseMixer block

Figure 2: The input with missing modality is harmonized through MoRe and PaKD, and then processed by PulseMixer to generate feature maps. These features are fused through UCPF to generate segmentation maps. "Available Modalities (Training Only)" indicates per-sample modality availability rather than privileged inputs.

3.2 Implicit Structural Alignment

MoRe is a lightweight projection with dual-path manifold expansion that decouples low-rank essential basis generation from structural-detail induction. To explicitly handle missing modalities, we introduce a binary indicator $b_m \in \{0, 1\}$, formulating the mapping for the m-th modality as:

Here, implicit structural alignment means that a recovered missing-modality representation approaches its real-modality reference while preserving crack-skeleton geometry and connectivity.

$$\mathbf{F}_m = b_m \cdot \mathcal{C}_{cat}(\Phi_{ifb}(\mathbf{X}_m),\, \Phi_{des}(\Phi_{ifb}(\mathbf{X}_m)))\,, \tag{1}$$

where $\mathcal{C}_{cat}$ is channel-wise concatenation, Φ_{ifb} is a linear projector, and Φ_{des} induces complementary details via depth-wise separable transforms; multi-scale coordinate attention then injects spatial positions into channels.

For adaptive harmonization, we construct uncertainty-driven dynamic manifold aggregation that separates confidence estimation from recalibration:

$$\boldsymbol{\alpha} = \mathrm{Softmax}\left(\left\{\mathbf{w}_2^\top \delta\big(\mathbf{w}_1 \mathrm{GAP}(\mathbf{F}_m)\big)\right\}_{m=1}^{M}\right), \tag{2}$$

$$\mathbf{F}_{latent} = \lambda_{res} \cdot \Psi_{calib}\left(\sum_{m=1}^{M} \alpha_m \mathbf{F}_m\right) + (1 - \lambda_{res}) \cdot \bar{\mathbf{F}}, \tag{3}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^M$ is the modality confidence vector derived from $\mathbf{w}_{1,2}$, SiLU δ, and Global feature $\mathrm{GAP}(\mathbf{F}_m)$. Since $\mathbf{F}_m$ is pre-multiplied by b_m, absent modalities contribute zero to the aggregation. $\Psi_{calib}(\cdot)$ is a nonlinear channel recalibration gate, and λ_{res} balances the recalibrated manifold with the modal mean $\bar{\mathbf{F}}$.

To equip the lightweight MoRe student with proactive crack topology awareness, we propose PaKD, using a frozen SAM2 as an implicit structural rule-maker.

Assuming topology is determined by a sparse keypoint set $\mathcal{K}$, we perform structure-aware sampling by decomposing cracks into topological subspaces $\mathcal{T} = \{\text{end}, \text{junction}, \text{skeleton}, \text{background}\}$ from $\mathbf{M}_{gt}$ and skeleton $\mathbf{S}_{skel}$, with prompts sampled independently:

$$\mathcal{P} = \bigcup_{\kappa \in \mathcal{T}} \left\{\mathbf{u}_i^{(\kappa)} \mid \mathbf{u}_i^{(\kappa)} \sim \mathcal{U}(\Omega_\kappa)\right\}_{i=1}^{N_\kappa}, \tag{4}$$

where Ω_κ is the candidate domain, $\mathcal{U}(\cdot)$ is uniform sampling, and N_κ controls structural diversity. We map $\mathcal{P}$ to a continuous Gaussian heatmap $\mathcal{H}_T$.

The prompt points are derived only from the existing training masks and require no additional endpoint, junction, or skeleton annotations. Neither validation nor test labels are accessed. SAM2, prompt sampling, and all distillation branches are removed at inference, so they introduce no deployment parameters, FLOPs, or latency.

To preserve semantic consistency, the distillation objective minimizes the deviation between teacher and student feature manifolds. The probability map distillation loss $\mathcal{L}_{prob}$ is defined as a pixel-wise Euclidean constraint:

$$\mathcal{L}_{prob} = \lambda_{MSE} \|\mathbf{P}_T - \mathbf{P}_S\|_2^2, \tag{5}$$

where $\mathbf{P}_{T/S}$ denote teacher and student probability maps. The squared L_2 norm $\|\cdot\|_2^2$ imposes strict pixel-level numerical consistency, balanced by the coefficient λ_{MSE}.

To inject topological priors missing from $\mathcal{L}prob$, we introduce a focal-style heatmap alignment loss $\mathcal{L}hm$ that emphasizes structurally sensitive regions. With $\delta_u = |\mathcal{H}_T(u) - \mathcal{H}_S(u)|$ and $\hat{p}_u = 1 - \delta_u$, the cross-entropy is modulated to reduce background dominance:

$$\mathcal{L}_{hm} = \mathbb{E}_{u\in\Omega}[-(1-\hat{p}_u)^{\gamma} \log(\hat{p}_u)], \tag{6}$$

where $\mathcal{H}_{T/S}(u)$ are normalized heatmap values over pixel set Ω. $\hat{p}_u$ quantifies local consistency confidence, and γ acts as a focusing factor to dynamically up-weight high-residual pixels, functioning as hard-example mining.

Furthermore, to prevent attention drift, a distributional inequality constraint is imposed, requiring the mean skeleton activation to exceed the background by ξ_s. Relaxing this constraint via a Hinge loss formulation yields the structural consistency loss $\mathcal{L}_{struc}$:

$$\mathcal{L}_{struc} = \left[\bar{\mathcal{H}}_S(\mathcal{R}_{bg}) - \bar{\mathcal{H}}_S(\mathcal{R}_{skel}) + \xi_s\right]_+ + (1 - \mathrm{Cov}(\mathbf{S}_{skel}, \mathcal{H}_S)), \tag{7}$$

where $[\cdot]_+ = \max(0, \cdot)$ is ReLU. $\bar{\mathcal{H}}_S(\mathcal{R}_{bg/skel})$ represent mean student activations over background and skeleton regions. ξ_s is the margin, and $\mathrm{Cov}(\cdot)$ aligns the activation distribution with the ground truth skeleton geometry $\mathbf{S}_{skel}$.

The contrastive loss enlarges the discriminative gap between foreground and background representations:

$$\mathcal{L}_{cont} = \sum_{c\in\{fg,bg\}} \left[\mu^{(c)}_{wrong} - \mu^{(c)}_{correct} + \xi_c\right]_+, \tag{8}$$

where $\mu^{(c)}_{correct/wrong}$ denote mean activations of features correctly/incorrectly classified into class c, with margin ξ_c.

These constraints enable Prism to internalize endpoint sensitivity and achieve full-scale implicit structural alignment.

3.3 Sparse Gated Pulse Propagation

PPI replaces raster scanning with a dynamic pulse sequence driven by skeletal geodesic distance on the latent manifold. Given the MoRe prior $\mathbf{F}_{latent}$, it predicts skeleton regions and simulates pulse diffusion from center to background to construct the fracture-manifold distance field $\mathcal{D}_{pulse}$:

$$\mathcal{D}_{pulse}(\mathbf{u}) = \min_{\mathbf{v}\in\Omega_{skel}} \|\mathbf{u} - \mathbf{v}\|_2, \tag{9}$$

where $\mathbf{u}$ is a pixel coordinate, $\mathbf{v}$ is a skeletal seed coordinate, Ω_{skel} is the skeletal seed set, and $\min_{\mathbf{v}\in\Omega_{skel}} \|\mathbf{u} - \mathbf{v}\|_2$ denotes the Euclidean distance from $\mathbf{u}$ to its nearest seed.

$$\Omega_{skel} = \{p \mid \sigma(\mathcal{F}_{skel}(\mathbf{F}_{latent}))_p > \tau\}, \tag{10}$$

where p indexes pixel locations, σ is Sigmoid, $\mathcal{F}_{skel}$ is the skeleton prediction head, $\mathbf{F}_{latent}$ is the latent prior feature, $(\cdot)_p$ denotes the value at pixel p, τ is the activation threshold, and Ω_{skel} collects activated skeletal seeds.

To enforce skeleton topology, we adopt a loss $\mathcal{L}_{ppi}$ combining binary cross-entropy and Dice terms:

$$\mathcal{L}_{ppi} = \mathcal{L}_{BCE}(\mathbf{S}_{map}, \mathbf{M}_{gt}) + \mathcal{L}_{Dice}(\mathbf{S}_{map}, \mathbf{M}_{gt}), \tag{11}$$

where $\mathbf{S}_{map} = \sigma(\mathcal{F}_{skel}(\mathbf{F}_{latent}))$ is the skeleton probability map, $\mathbf{M}_{gt}$ is the GT mask, $\mathcal{L}_{BCE}(\cdot,\cdot)$ is the binary cross-entropy term, and $\mathcal{L}_{Dice}(\cdot,\cdot)$ is the Dice loss term.

$\mathcal{I}_{pulse}$ follows a gradient-descent ordering, using a small spatial perturbation to resolve equidistant ambiguities:

$$\mathcal{I}_{pulse} = \underset{u \in \Omega}{\text{argsort}} \left(\mathcal{D}_{pulse}(u) + \epsilon \cdot \mathcal{N}(u) \right), \tag{12}$$

where Ω is the image domain, u indexes pixels in Ω, argsort returns the ascending index order, ϵ is the perturbation coefficient, and $\mathcal{N}(u)$ is a spatial noise term.

Using index mapping $\tilde{\mathbf{X}} = \text{Gather}(\mathbf{X}, \mathcal{I}_{pulse})$, features are reordered as skeleton, boundary, and background, and a WKV linear recurrence aggregates historical information along this sequence:

$$\mathbf{wkv}_t = \frac{\sum_{i=1}^{t-1} e^{-(t-1-i)w+\mathbf{k}_i} \odot \mathbf{v}_i + e^{u+\mathbf{k}_t} \odot \mathbf{v}_t}{\sum_{i=1}^{t-1} e^{-(t-1-i)w+\mathbf{k}_i} + e^{u+\mathbf{k}_t}}, \tag{13}$$

where $\mathbf{wkv}_t$ is the aggregated output at step t, t is the current index, i indexes historical steps, w is the decay parameter, u is the current gain, $\mathbf{k}_i$ and $\mathbf{k}_t$ are key vectors at steps i and t, $\mathbf{v}_i$ and $\mathbf{v}_t$ are value vectors at steps i and t, $\odot$ denotes element-wise multiplication.

Standard linear recurrences exhibit a static parameterization bottleneck, since $\mathbf{wkv}_t$ is generated by a fixed linear combination of historical states. We introduce sparse gating to transform this static mapping into a dynamic, input-conditioned modulation:

$$\mathbf{W}_{dyn}(\mathbf{X}) = \text{diag}(\mathcal{G}_{sparse}(\mathbf{X})) \cdot \mathbf{W}_{wkv}, \tag{14}$$

$$\dim\left(\text{span}\left(\{\mathbf{W}_{dyn}(\mathbf{X})\}_{\mathbf{X} \in \mathcal{D}}\right)\right) \gg \text{rank}(\mathbf{W}_{wkv}), \tag{15}$$

where $\mathbf{W}_{dyn}(\mathbf{X})$ is the effective dynamic transition matrix formed by modulating the static base $\mathbf{W}_{wkv}$ with the data-dependent gate $\mathcal{G}_{sparse}$. This input-specific adaptation breaks the static parameterization bottleneck, thereby significantly expanding the effective linear span over the entire dataset $\mathcal{D}$ to approximate highly non-linear manifolds.

Based on this, the data-dependent gate $\mathcal{G}_{sparse}$ explicitly modulates the WKV aggregate $\mathbf{wkv}_t$ via element-wise multiplication to suppress low-confidence responses:

$$\mathbf{Y}_t = \mathbf{wkv}_t \odot \left[\sigma(\mathbf{W}_g \mathbf{X}_t + \mathbf{b}_g) - \theta\right]_+, \tag{16}$$

where $\mathbf{Y}_t$ is the final gated output, $[\cdot]_+$ denotes ReLU, $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable parameters, and θ is a learnable threshold. A layer-aware factor α_l further modulates gating intensity across layers.

To reinforce geometric continuity, the Geometric Propagation Module (GPM) combines adaptive strip convolutions with frequency-domain enhancement and formulates topology-aware diffusion as:

$$\mathbf{F}_{diff} = \mathcal{G}_{diff}(\mathbf{F}_{fuse}) \odot \mathcal{K}_{prop}(\mathbf{F}_{fuse}) + (1 - \mathcal{G}_{diff}(\mathbf{F}_{fuse})) \odot \mathcal{H}_{edge}(\mathbf{F}_{fuse}), \tag{17}$$

where $\mathbf{F}_{fuse}$ is the intermediate representation, $\mathcal{K}_{prop}$ is a large-kernel convolution, $\mathcal{H}_{edge}$ is the edge-preservation term, and $\mathcal{G}_{diff}$ is the dynamic diffusion gate, which propagates features from high-response regions to low-response breakpoints while preserving edge details.

3.4 Uncertainty-Guided Connectivity Fusion

UCPF utilizes pixel-level uncertainty estimation to suppress noise and drive feature flow from high-confidence regions to low-confidence breakpoints.

Specifically, a lightweight head $\mathcal{H}_{unc}$ predicts pixel-level uncertainty for each modal feature $\mathbf{F}_m$, yielding normalized confidence weights $\mathbf{w}_m$:

$$\mathbf{w}_m = \frac{1 - \sigma(\mathcal{H}_{unc}(\mathbf{F}_m))}{\sum_{k=1}^{M} (1 - \sigma(\mathcal{H}_{unc}(\mathbf{F}_k)))} + \epsilon, \tag{18}$$

where σ is the Sigmoid function, ϵ is a stability term, and Ψ_{se} used for channel-wise feature recalibration. The fused representation is refined as $\mathbf{F}_{fused} = \Psi_{se}(\sum_{m=1}^{M} \mathbf{w}_m \odot \mathbf{F}_m)$, ensuring high-reliability modalities dominate.

A confidence-gated propagation adopts a pre-weighted propagation strategy, defining the propagation flow $\mathbf{F}_{flow}$ as:

$$\mathbf{F}_{flow} = \mathcal{K}_{prop}\left(\mathbf{F}_{fused} \odot \sigma(\mathcal{H}_{conf}(\mathbf{F}_{fused}))\right), \tag{19}$$

where $\mathcal{H}_{conf}$ is a confidence head and $\mathcal{K}_{prop}$ is a large-kernel grouped convolution.

Finally, a gated residual produces the decoded feature $\mathbf{F}_{rec}$:

$$\mathbf{F}_{rec} = \text{BN}(\mathbf{F}_{fused} + \sigma(\mathcal{H}_{gate}(\mathbf{F}_{flow})) \odot \mathbf{F}_{flow}), \tag{20}$$

where $\mathcal{H}_{gate}$ is a gated convolution; residual connections preserve details while $\mathbf{F}_{flow}$ fills topological gaps, followed by a decoder to produce the final segmentation.

Table 2: Performance comparison under dual-modal and multi missing settings. **Purple** and Green denote best and second-best results.

	Missing Ratio		CMX		PrimKD		Sigma		CMNeXT		MCubeS		GminiFu		PWRF		LIDAR		**Prism (Ours)**	
	RGB	Infrared	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
IRTCrack	10%	Full	0.8330	0.8414	0.8161	0.8327	0.8327	0.8344	0.8224	0.8311	0.8212	0.8378	0.7718	0.7885	0.8003	0.8224	0.8036	0.8235	**0.8469**	**0.8456**
	30%	Full	0.7628	0.8143	0.7720	0.8054	0.7892	0.8040	0.7809	0.8116	0.7828	0.8021	0.7002	0.7673	0.7573	0.7977	0.7722	0.7847	**0.8106**	**0.8352**
	50%	Full	0.6960	0.7750	0.7193	0.7639	0.7238	0.7749	0.7371	0.7837	0.7003	0.7650	0.6093	0.7184	0.7048	0.7740	0.7083	0.7676	**0.7789**	**0.8048**
	70%	Full	0.6766	0.7257	0.6638	0.7384	0.6942	0.7376	0.6408	0.7289	0.6672	0.7448	0.5509	0.6976	0.6536	0.7421	0.6516	0.7200	**0.7035**	**0.7651**
	90%	Full	0.5929	0.6940	0.5795	0.6932	0.6088	0.7023	0.6186	0.7081	0.6000	0.7028	0.5082	0.6488	0.6151	0.6958	0.6209	0.6956	**0.6553**	**0.7300**
	Full	10%	0.8387	0.8458	0.8146	0.8389	0.8426	0.8477	0.8302	0.8481	0.8107	0.8424	0.7911	0.8237	0.8120	0.8381	0.8234	0.8280	**0.8666**	**0.8631**
	Full	30%	0.8315	0.8477	0.7931	0.8306	0.8328	0.8341	0.8286	0.8452	0.8211	0.8431	0.7877	0.8214	0.8060	0.8351	0.8197	0.8249	**0.8569**	**0.8587**
	Full	50%	0.8289	0.8441	0.7946	0.8298	0.8131	0.8284	0.8265	0.8388	0.8145	0.8282	0.7664	0.8103	0.7949	0.8292	0.8115	0.8165	**0.8578**	**0.8591**
	Full	70%	0.8327	0.8456	0.7905	0.8278	0.8021	0.8130	0.8156	0.8399	0.8109	0.8278	0.7654	0.7996	0.7972	0.8274	0.8065	0.8137	**0.8491**	**0.8524**
	Full	90%	0.8245	0.8437	0.7879	0.8306	0.7995	0.8240	0.8022	0.8210	0.7954	0.8222	0.7609	0.7980	0.7715	0.8056	0.7877	0.7969	**0.8493**	**0.8478**
	Full	Full	0.8463	0.8518	0.8331	0.8447	0.8576	0.8509	0.8330	0.8466	0.8352	0.8504	0.8108	0.8214	0.8203	0.8383	0.8580	0.8538	**0.8674**	**0.8636**
	RGB	Depth	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
CrackDepth	10%	Full	0.7127	0.7890	0.7770	0.8235	0.7274	0.7838	0.7705	0.8178	0.7736	0.8215	0.6988	0.7787	0.7448	0.8050	0.7762	0.8164	**0.8136**	**0.8341**
	30%	Full	0.7039	0.7658	0.7077	0.7738	0.7164	0.7812	0.7166	0.7856	0.6918	0.7649	0.6471	0.7553	0.6876	0.7668	0.6646	0.7338	**0.7459**	**0.8068**
	50%	Full	0.6475	0.7485	0.6494	0.7498	0.5475	0.7364	0.6248	0.7437	0.6222	0.7215	0.5303	0.7024	0.6198	0.7312	0.5424	0.7188	**0.6729**	**0.7596**
	70%	Full	0.5439	0.6914	0.5348	0.6886	0.4995	0.6744	0.5139	0.6712	0.5103	0.6805	0.1668	0.5364	0.5348	0.6808	0.4557	0.6170	**0.5929**	**0.7179**
	90%	Full	0.4304	0.6355	0.0641	0.3472	0.0570	0.0584	0.4046	0.6227	0.4142	0.6394	0.1095	0.5136	0.4046	0.6233	0.2445	0.5526	**0.5361**	**0.6847**
	Full	10%	0.8003	0.8408	0.7944	0.8363	0.8082	0.8426	0.7953	0.8369	0.7950	0.8384	0.7519	0.8218	0.7682	0.8198	0.7915	0.8292	**0.8298**	**0.8497**
	Full	30%	0.7939	0.8386	0.7905	0.8299	0.7971	0.8380	0.7854	0.8286	0.7819	0.8350	0.7570	0.8175	0.7603	0.8166	0.7802	0.7924	**0.8287**	**0.8486**
	Full	50%	0.7881	0.8367	0.7862	0.8294	0.7929	0.8291	0.7848	0.8257	0.7753	0.8203	0.6889	0.7984	0.7497	0.8126	0.7607	0.8114	**0.8249**	**0.8481**
	Full	70%	0.7896	0.8336	0.7894	0.8300	0.7855	0.8287	0.7641	0.8165	0.7702	0.8158	0.6224	0.7306	0.7481	0.8098	0.7570	0.8029	**0.8205**	**0.8456**
	Full	90%	0.7712	0.8283	0.7733	0.8278	0.7517	0.8184	0.7482	0.8157	0.7431	0.8089	0.5482	0.7025	0.7203	0.8023	0.7276	0.7861	**0.8217**	**0.8478**
	Full	Full	0.8109	0.8410	0.7989	0.8338	0.8172	0.8445	0.8043	0.8407	0.8071	0.8412	0.7887	0.8261	0.7795	0.8210	0.8180	0.8453	**0.8356**	**0.8537**

3.5 Objective Function

To ensure segmentation accuracy while achieving effective latent structural alignment and topological connectivity preservation, a unified objective function is designed:

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \mathcal{L}_{prob} + \mathcal{L}_{pakd} + \mathcal{L}_{ppi}, \tag{21}$$

where $\mathcal{L}_{seg} = \mathcal{L}_{BCE} + \mathcal{L}_{Dice}$ enforces semantic consistency of the final prediction [12, 38]; $\mathcal{L}_{prob}$ aligns MoRe and SAM2 probability maps; $\mathcal{L}_{pakd} = \mathcal{L}_{hm} + \mathcal{L}_{struc} + \mathcal{L}_{cont}$ aggregates PaKD heatmap, structural, and contrastive losses; and $\mathcal{L}_{ppi}$ supervises PPI skeleton prediction for pulse sequences.

4 Experiments

4.1 Datasets

CrackDepth [25]: Includes 656 Lytro Illum RGB-D pairs across four scenarios. The depth modality captures geometric fluctuations, robustly characterizing crack edges under complex illumination.

CrackPolar [25]: Contains 986 samples captured by a DAHENG MERCURY2 camera, including RGB images and Stokes parameters, leveraging stress polarization correlations to reveal microscopic crack details.

IRTCrack [21]: Comprises 448 RGB-Thermal pairs captured by a FLUKE TiX580, where thermal data reveals subsurface anomalies.

Table 1: Complexity comparison of dual-modal 512 × 512 input.

Method	Year	FLOPs↓	Params↓	Size↓
MCubeS [19]	CVPR'22	165.59G	117.76M	2253MB
CMX [53]	TITS'23	56.99G	66.56M	762MB
CMNeXT [54]	CVPR'23	44.76G	57.63M	660MB
PrimKD [11]	MM'24	679.09G	615.64M	7322MB
GminiFu [14]	ICML'24	45.53G	29.19M	472MB
PWRF [29]	IJCV'25	185.14G	252.36M	1024MB
Sigma [40]	WACV'25	63.43G	41.68M	553MB
LIDAR [25]	MM'25	33.33G	5.35M	78MB
Prism	N/A	**30.68G**	**2.62M**	**38MB**

4.2 Implementation Details

We implemented Prism using PyTorch V2.9.1 on a server equipped with an Intel Xeon Platinum 8558 CPU and an NVIDIA L20 GPU, running Ubuntu 22.04.5. Training was conducted using the AdamW optimizer with an initial learning rate of 2×10^{-3} and a weight decay of 0.01. A PolyLR scheduler was employed over 100 epochs. All baselines were retrained from scratch under the same stochastic missingness protocol, modality-specific missing ratios, and fixed 7:1:2 split as Prism; no baseline was trained only on complete inputs.

4.3 Comparison with SOTA Methods

Comparative Experiments on Dual-modal Input: As shown in Table 2 and Figure 3, we conducted extensive comparisons on the dual-modal datasets IRTCrack [21] and CrackDepth [25].

On IRTCrack, Prism consistently outperforms CMX [53], at 70% RGB missing it improves F1 by 3.98% and mIoU by 5.42%, and at 30% Infrared missing by 3.06% and 1.30%. This robustness across missing rates stems from MoRe's implicit structural alignment and UCPF's uncertainty-aware fusion, which recover sub-surface thermal anomalies despite missing textures. On CrackDepth, Prism further demonstrates robustness, achieving relative gains of 119.26% in F1 and 23.90% in mIoU over

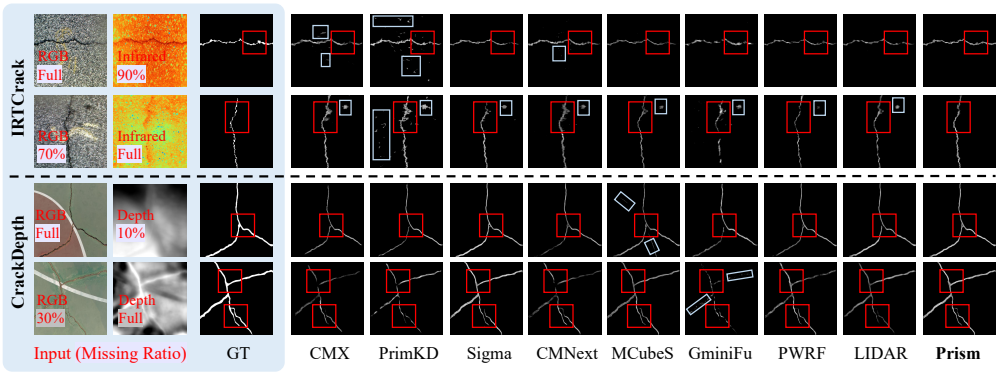


Figure 3: Visual comparison under dual-modal multi missing settings. Red boxes denote key regions; blue denote misidentifications.

Table 3: Performance comparison under three-modal and multi missing settings. **Purple** and Green denote best and second-best results.

Missing Ratio			GminiFusion		MCubeS		mmsFormer		PWRF		CMNeXT		LIDAR		**Prism (Ours)**	
RGB	AoP	DoP	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
Full	10%	30%	0.6766	0.7706	0.7250	0.7955	0.7191	0.7912	0.7056	0.7823	0.7288	0.7961	0.7179	0.7891	**0.7427**	**0.8038**
30%	Full	10%	0.5836	0.7012	0.7090	0.7820	0.7056	0.7896	0.6904	0.7782	0.7091	0.7889	0.6948	0.7708	**0.7354**	**0.7997**
10%	30%	Full	0.6997	0.7816	0.7127	0.7918	0.7104	0.7904	0.7035	0.7794	0.7154	0.7913	0.6903	0.7549	**0.7451**	**0.8014**
Full	70%	90%	0.6699	0.7653	0.6962	0.7900	0.7043	0.7897	0.6927	0.7804	0.7111	0.7927	0.6984	0.7784	**0.7392**	**0.8015**
90%	Full	70%	0.3003	0.5927	0.6705	0.7687	0.6410	0.7490	0.6078	0.7282	0.6795	0.7691	0.5964	0.7165	**0.7267**	**0.7930**
70%	90%	Full	0.5412	0.7051	0.5566	0.7002	0.6410	0.7490	0.6404	0.7373	0.5372	0.6897	0.5846	0.7135	**0.6886**	**0.7745**
10%	30%	70%	0.6187	0.7439	0.7018	0.7885	0.6834	0.7628	0.6670	0.7704	0.6988	0.7805	0.6868	0.7622	**0.7394**	**0.8012**
30%	70%	10%	0.5728	0.7163	0.6778	0.7747	0.6696	0.7692	0.6290	0.7493	0.6935	0.7795	0.5810	0.6978	**0.7285**	**0.7924**
70%	10%	30%	0.4041	0.6722	0.6812	0.7743	0.6899	0.7721	0.6612	0.7677	0.6900	0.7788	0.6261	0.7077	**0.7229**	**0.7891**
30%	70%	90%	0.1938	0.6302	0.5818	0.7352	0.5408	0.7234	0.2025	0.6679	0.5731	0.7153	0.4555	0.7034	**0.6619**	**0.7523**
70%	90%	30%	0.2249	0.5682	0.5476	0.6970	0.1958	0.6503	0.2226	0.6585	0.4863	0.6965	0.4164	0.6243	**0.6244**	**0.7235**
90%	30%	70%	0.5001	0.6754	0.5827	0.7100	0.5066	0.6690	0.2207	0.6649	0.5922	0.7130	0.3716	0.5885	**0.6307**	**0.7315**
Full	Full	Full	0.6920	0.7794	0.7163	0.7936	0.7265	0.7952	0.7063	0.7837	0.7124	0.7875	0.7353	0.7976	**0.7510**	**0.8066**

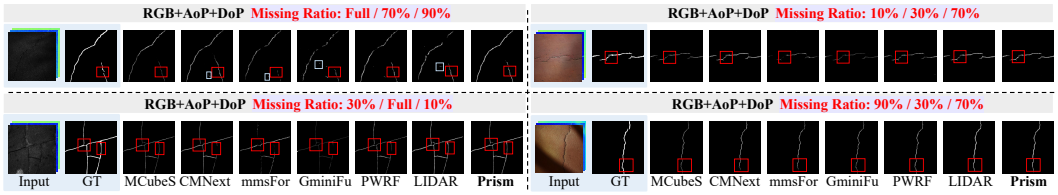


Figure 4: Visual comparison under three-modal multi missing settings. Red boxes denote key regions; blue denote misidentifications.

LIDAR [25] at 90% RGB missing, as PulseMixer leverages depth geometry and spatial hierarchy when RGB fails; even with full modalities, Prism still exceeds LIDAR by 2.14% in F1.

As shown in Table 1, Prism incurs the lowest overhead at a 512×512 resolution. Compared to LIDAR [25], it reduces Model Size by 51.28%, Params by 51.03%, and FLOPs by 7.77%; relative to PrimKD [11], it reduces parameter count by 235 times and model size by 193 times. This efficiency comes from PulseMixer sparse gating that compresses feature redundancy and the lightweight designs of MoRe and PulseMixer. With only four Blocks, Prism yields topology-clear segmentation with minimal background interference, maintaining linear computational complexity and a performance efficiency balance.

Table 4: Complexity comparison of three-modal 512 × 512 input.

Method	Year	FLOPs↓	Params↓	Size↓
MCubeS [19]	CVPR 2022	242.65G	176.29M	2867MB
CMNeXT [54]	CVPR 2023	45.99G	57.64M	693MB
mmsFormer [36]	OJSP 2024	59.28G	33.31M	435MB
GminiFu [14]	ICML 2024	46.79G	29.22M	494MB
PWRF [29]	IJCV 2025	238.42G	344.02M	3789MB
LIDAR [25]	MM 2025	50.00G	8.02M	150MB
Prism (Ours)	N/A	**44.77G**	**3.86M**	**56MB**

Comparative Experiments on Three-modal Input: We conducted extensive experiments on CrackPolar [25] with RGB, AoP, and DoP modalities. As shown in the Table 4 and Figure 4, Prism achieves the best performance under all missing-ratio settings.

With RGB complete and AoP and DoP missing at 10% and 30%, Prism achieves an F1 of 0.7427 and an mIoU of 0.8038, surpassing CMNeXT [54] by 1.90% and 0.96%, indicating that implicit structural alignment registers incomplete polarization cues with complete RGB features for high precision fusion. In the extreme setting where only DoP is complete, with RGB missing at 70% and AoP missing at 90%, Prism still attains an F1 of 0.6886 and an mIoU of 0.7745, producing clear segmentation with suppressed noise under low illumination and complex textures, suggesting that MoRe and PaKD align missing modality manifolds toward complete DoP representations to reduce data-scarcity degradation.

When all three modalities are partially missing, for example RGB missing at 10%, AoP missing at 30%, and DoP missing at 70%, Prism exceeds MCubeS [19] by 5.35% in F1. At 90% RGB, 30% AoP, and 70% DoP missing, Prism achieves an F1 of 0.6307 and an mIoU of 0.7315, exceeding LIDAR [25] by 69.73% and 24.29%, showing that MoRe, PulseMixer, and UCPF preserve physical attributes and geometric structure even when the primary modality fails.

As shown in Table 4, Prism retains exceptionally low overhead for three-modality inputs at a 512×512 resolution, requiring 44.77G FLOPs, 3.86M parameters, and a 56MB model size, which are 10.34%, 51.87%, and 62.67% lower than LIDAR [25], respectively. This efficiency further validates PulseMixer sparse gating and MoRe parameter efficient design for multimodal processing, reducing computational redundancy and enabling practical deployment on edge devices.

4.4 Ablation Study

We conducted ablation studies on CrackPolar [25] using RGB, AoP, and DoP modalities. The RGB modality was fully available, while AoP and DoP were each set to a 50% missing rate.

Ablation Studies within PaKD: To evaluate PaKD, we conduct ablations, as shown in Figure 14. Any distillation improves over the no-distillation baseline, with full PaKD performing best and yielding relative gains of 2.23% in F1 and 0.83% in mIoU, confirming the value of transferring SAM2's implicit structural knowledge. Probability map original distillation alone provides limited gains and is clearly inferior to PaKD; PaKD improves F1 by 1.16% and mIoU by 0.78%, indicating that pixel-level alignment cannot convey complex crack topology, which PaKD enforces via explicit structural constraints. Further analysis show that removing the heatmap alignment, structural consistency, or contrastive loss degrades performance; specifically, removing heatmap alignment reduces F1 by 0.96% and mIoU by 0.47% due to weaker skeleton localization, and removing contrastive loss reduces F1 by 0.84%, reflecting reduced separation between foreground and background.

Table 5: Comparison of different missing modality processing strategies.

Method	F1	mIoU	Params↓	FLOPs↓	Size↓
KB [15]	0.7296	0.7984	4.18M	45.95G	57MB
MCMoE [47]	0.7319	0.7992	4.06M	45.96G	57MB
TouchFormer [31]	0.7219	0.7944	6.99M	45.95G	90MB
MD2N [7]	0.7135	0.7889	12.03M	45.96G	85MB
CCSD [45]	0.7357	0.8014	3.90M	102.55G	57MB
SKD [37]	0.7337	0.8013	16.09M	50.01G	180MB
Prism (Ours)	**0.7443**	**0.8049**	**3.86M**	**44.77G**	**56MB**

Ablation Studies on Different Missing Modality Handling Methods: As shown in Table 5, Prism outperforms non-distillation methods including KB [15], TouchFormer [31], MD2N [7], and MCMoE [47] by surpassing the top-performing MCMoE by 1.69% in F1. This stems from PaKD, enabling MoRe to inject SAM2's prompt-driven topological priors to overcome input bottlenecks. Compared to distillation baselines, Prism surpasses CCSD [45] by 1.17% in F1 and 0.44% in mIoU. Distinct from SKD utilizing spectral alignment or CCSD relying on hierarchical distribution matching, PaKD distills prompt sensitivity into MoRe to reconstruct missing details via manifold continuity. Furthermore, Prism redefines efficiency as the lightweight MoRe requires only 3.86M parameters and 44.77G FLOPs unlike the heavier SKD or the 102.55G FLOPs of CCSD. This 56.28% reduction in FLOPs confirms that implicit structural alignment achieves superior topological preservation with significantly higher efficiency.

Table 6: Performance comparison of different insight strategies.

Insight Method	F1	mIoU	Delay Time↓
Para	0.7251	0.7978	1.63E-03s
Para_Snake	0.7334	0.7949	2.21E-03s
Para_Snake_Bi	0.7335	0.8007	1.55E-03s
Diag	0.7363	0.7988	2.17E-03s
Diag_Snake	0.7340	0.7981	2.40E-03s
Diag_Snake_Bi	0.7349	0.7973	6.25E-04s
SASS [24]	0.7384	0.8008	2.16E-03s
EDG [25]	0.7392	0.7983	**7.15E-07s**
Q-Shift [8]	0.7290	0.8014	9.82E-05s
PPI (Ours)	**0.7443**	**0.8049**	8.32E-05s

Ablation Studies on Different Insight Strategies: Table 6 compares PPI with controlled parallel, bidirectional, snake, and diagonal scan schedules derived from representative state-space designs [57, 30, 58, 61], together with SASS [24], EDG [25], and Q-Shift [8]; "Para" and "Snake" are labels for our implemented schedules rather than official method names. With PPI, Prism achieves the best F1 and mIoU, exceeding EDG by 0.69% and 0.83% and surpassing SASS by 0.80% and 0.50%,

Table 7: Targeted evaluation of output structure and recovered feature fidelity. Blue bold and green underlined entries denote best and second-best results.

	Output Structure (RGB Full; AoP and DoP 50% Missing)				Recovered DoP (DoP Hidden)			
Method	clDice↑	Skeleton-F1↑	Boundary-F1↑	Component Error↓	nMSE↓	MAE↓	Cosine↑	Linear CKA↑
MD2N	0.8440	0.7831	0.7652	1.5303	0.3379	0.2261	0.8279	<u>0.1287</u>
MCMoE	<u>0.8646</u>	<u>0.7950</u>	<u>0.7771</u>	<u>1.4385</u>	<u>0.2169</u>	0.1831	<u>0.8907</u>	0.0387
SKD	0.8568	0.7882	0.7742	1.7593	0.2329	<u>0.1719</u>	0.8814	0.0771
Prism	**0.8726**	**0.8122**	**0.7910**	**1.3378**	**0.1862**	**0.1552**	**0.9056**	**0.2353**

indicating that side supervised progressive insights improve cross-modal crack topology modeling. Prism also outperforms Q-Shift by 2.10% in F1 and 0.44% in mIoU, showing the advantage of skeleton-driven gradual guidance over feature shifting. PPI further processes each image set in 8.32E-05s, which is 1.18× faster than Q-Shift and about 26× faster than SASS, enabling preset-free gains in both accuracy and efficiency without pre-training or manual intervention for edge and real-time deployment.

4.5 Targeted Structural Evaluation

Beyond global F1 and mIoU, we evaluate implicit structural alignment on CrackPolar with RGB fully available and both AoP and DoP subject to a 50% missing rate. All output metrics are averaged per image. Geometry is measured by clDice, Skeleton-F1, and Boundary-F1 with a two-pixel tolerance, while connectivity is measured by the absolute difference in the number of 8-connected components normalized by the ground-truth count. For feature fidelity, DoP is hidden while RGB and AoP remain available; all methods use the same normalization and common feature space, and real DoP encoded by the same frozen encoder is used only as an evaluation reference.

As shown in Table 7, Prism achieves the best result for every metric. It improves clDice, Skeleton-F1, and Boundary-F1 by 0.80, 1.72, and 1.39 percentage points over the strongest respective baselines, while reducing normalized component error by 7.0%. Before decoding, Prism also reduces nMSE from 0.2169 to 0.1862 and MAE from 0.1719 to 0.1552, while increasing cosine similarity and Linear CKA to 0.9056 and 0.2353. These results independently support both the output-level topology preservation and latent fidelity of MoRe. Additional prompt, failure-mode, and serialization diagnostics are provided in the appendix.

5 Conclusion

We propose Prism, a lightweight pulse RWKV framework for multimodal structural crack segmentation under diverse missing-modality settings. By integrating MoRe, PaKD, PulseMixer, and UCPF, Prism harmonizes missing modalities from available feature manifolds, distills prompt-driven structural sensitivity, models anisotropic crack topology with skeletal pulse propagation, and performs uncertainty-aware fusion to produce clean and detailed segmentation maps. Experiments on three datasets demonstrate SOTA performance across various missing-modality scenarios with extremely low complexity; notably, on the light-field depth dataset with a 90% depth missing rate, Prism achieves an F1 of 0.8217 and an mIoU of 0.8478 with only 2.62M parameters, highlighting its suitability for resource-constrained industrial deployment.

Limitations. Although Prism remains lightweight during inference, there is still room to simplify its training pipeline, where PaKD leverages a frozen teacher and structural objectives for implicit guidance. Future work will investigate more streamlined supervision and explicit texture generation to further improve physical consistency and detail recovery.

Acknowledgments and Disclosures

Acknowledgments. This work was supported by the National Natural Science Foundation of China (NSFC) under Grants T2422015, and 62306212; the Beijing-Tianjin-Hebei Natural Science Foundation Cooperation Project under Grant 25JJJC0009; the China Postdoctoral Science Foundation under Grant 2026M791667;and the Postdoctoral Fellowship Program of the China Postdoctoral Science Foundation (CPSF) under Grant GZC20260889.

Competing Interests. The authors declare no competing interests.

References

[1] Dariush Amirkhani, Mohand Saïd Allili, and Jean-François Lapointe. Cracksight: An efficient crack segmentation model in varying acquisition ranges and complex backgrounds. *IEEE Transactions on Automation Science and Engineering*, 22:19197–19214, 2025.

[2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, pages 348–367. Springer, 2022.

[3] Linwei Chen, Ying Fu, Lin Gu, Chenggang Yan, Tatsuya Harada, and Gao Huang. Frequency-aware feature fusion for dense image prediction. *IEEE transactions on pattern analysis and machine intelligence*, 46(12):10763–10780, 2024.

[4] Tianxiang Chen, Xudong Zhou, Zhentao Tan, Yue Wu, Ziyang Wang, Zi Ye, Tao Gong, Qi Chu, Nenghai Yu, and Le Lu. Zig-rir: Zigzag rwkv-in-rwkv for efficient medical image segmentation. *IEEE Transactions on Medical Imaging*, 44(8):3245–3257, 2025.

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

[6] Xu Cheng, Tian He, Fan Shi, Meng Zhao, Xiufeng Liu, and Shengyong Chen. Selective feature fusion and irregular-aware network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):3445–3456, 2024.

[7] Ruiting Dai, Chenxi Li, Yandong Yan, Lisi Mo, Ke Qin, and Tao He. Unbiased missing-modality multimodal learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24507–24517, 2025.

[8] Yuchen Duan, Weiyun Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, Hongsheng Li, Jifeng Dai, and Wenhai Wang. Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures. In *The Thirteenth International Conference on Learning Representations*, 2025.

[9] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *European Conference on Computer Vision*, pages 222–241. Springer, 2024.

[10] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28124–28133, 2025.

[11] Zhiwei Hao, Zhongyu Xiao, Yong Luo, Jianyuan Guo, Jing Wang, Li Shen, and Han Hu. Primkd: Primary modality guided multimodal fusion for rgb-d semantic segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1943–1951, 2024.

[12] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE conference on computational intelligence in bioinformatics and computational biology*, pages 1–7. IEEE, 2020.

[13] Achref Jaziri, Martin Mundt, Andres Fernandez, and Visvanathan Ramesh. Designing a hybrid neural system to learn real-world crack segmentation from fractal-based simulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8636–8646, 2024.

[14] Ding Jia, Jianyuan Guo, Kai Han, Han Wu, Chao Zhang, Chang Xu, and Xinghao Chen. Geminifusion: Efficient pixel-wise multimodal fusion for vision transformer. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 21753–21767. PMLR, 2024.

[15] Guanzhou Ke, Shengfeng He, Xiaoli Wang, Bo Wang, Guoqing Chao, Yuanyang Zhang, Yi Xie, and Hexing Su. Knowledge bridger: Towards training-free missing modality completion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25864–25873, 2025.

[16] Yi-Lun Lee, Yi-Hsuan Tsai, Wei-Chen Chiu, and Chen-Yu Lee. Multimodal prompting with missing modalities for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14943–14952, 2023.

[17] Bingyu Li, Da Zhang, Zhiyuan Zhao, Junyu Gao, and Xuelong Li. Stitchfusion: Weaving any visual modalities to enhance multimodal semantic segmentation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1308–1317, 2025.

[18] Sijie Li, Chen Chen, and Jungong Han. Simmlm: A simple framework for multi-modal learning with missing modality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24068–24077, 2025.

[19] Yupeng Liang, Ryosuke Wakaki, Shohei Nobuhara, and Ko Nishino. Multimodal material segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19800–19808, 2022.

[20] Jianghai Liao, Yuanhao Yue, Dejin Zhang, Wei Tu, Rui Cao, Qin Zou, and Qingquan Li. Automatic tunnel crack inspection using an efficient mobile imaging module and a lightweight cnn. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15190–15203, 2022.

[21] Fangyu Liu, Jian Liu, and Linbing Wang. Asphalt pavement crack detection based on convolutional neural network and infrared thermography. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22145–22155, 2022.

[22] Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. Crackformer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3783–3792, 2021.

[23] Huajun Liu, Jing Yang, Xiangyu Miao, Christoph Mertz, and Hui Kong. Crackformer network for pavement crack segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 24 (9):9240–9252, 2023.

[24] Hui Liu, Chen Jia, Fan Shi, Xu Cheng, and Shengyong Chen. Scsegamba: lightweight structure-aware vision mamba for crack segmentation in structures. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29406–29416, 2025.

[25] Hui Liu, Chen Jia, Fan Shi, Xu Cheng, Mengfei Shi, Xia Xie, and Shengyong Chen. Lidar: Lightweight adaptive cue-aware fusion vision mamba for multimodal segmentation of structural cracks. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1832–1841, 2025.

[26] Mushui Liu, Jun Dan, Ziqian Lu, Yunlong Yu, Yingming Li, and Xi Li. CM-UNet: Hybrid CNN-Mamba UNet for remote sensing image semantic segmentation. *arXiv preprint arXiv:2405.10530*, 2024.

[27] Xiaohao Liu, Xiaobo Xia, Jiaheng Wei, Shuo Yang, Xiu Su, See-Kiong Ng, and Tat-Seng Chua. Calibrated multimodal representation learning with missing modalities. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 79140–79160. PMLR, 2026.

[28] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14420–14430, 2023.

[29] Yi Liu, Chengxin Li, Shoukun Xu, and Jungong Han. Part-whole relational fusion towards multi-modal scene understanding. *International Journal of Computer Vision*, 133(7):4483–4503, 2025.

[30] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

[31] Kailin Lyu, Long Xiao, Jianing Zeng, Junhao Dong, Xuexin Liu, Zhuojun Zou, Haoyue Yang, Lin Shu, and Jie Hao. Touchformer: A robust transformer-based framework for multimodal material perception. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22): 18496–18504, 2026.

[32] Junlin Ouyang, Qingwang Wang, Ying Shang, Pengcheng Jin, Hangwei Zhong, Liman Zhou, and Tao Shen. An rgb-tir dataset from uav platform for robust urban traffic scenes semantic segmentation. *Scientific Data*, 12(1):1701, 2025.

[33] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

[34] Danfeng Qin, Chas Leichner, Manolis Delakis, Marco Fornoni, Shixin Luo, Fan Yang, Weijun Wang, Colby R. Banbury, Chengxi Ye, Berkin Akin, Vaibhav Aggarwal, Tenghui Zhu, Daniele Moro, and Andrew G. Howard. MobileNetV4: Universal models for the mobile ecosystem. In *Computer Vision – ECCV 2024*, volume 15098 of *Lecture Notes in Computer Science*, pages 78–96. Springer, 2024.

[35] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Md Kaykobad Reza, Ashley Prater-Bennette, and M Salman Asif. Mmsformer: Multimodal transformer for material and semantic segmentation. *IEEE Open Journal of Signal Processing*, 5:599–610, 2024.

[37] Aniruddh Sikdar, Jayant Teotia, and Suresh Sundaram. Skd-net: Spectral-based knowledge distillation in low-light thermal imagery for robotic perception. In *2024 IEEE International Conference on Robotics and Automation*, pages 9041–9047. IEEE, 2024.

[38] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, volume 10553 of *Lecture Notes in Computer Science*, pages 240–248. Springer, 2017.

[39] Zhengzheng Tu, Tian Xia, Chenglong Li, Xiaoxiao Wang, Yan Ma, and Jin Tang. RGB-T image saliency detection via collaborative graph learning. *IEEE Transactions on Multimedia*, 22(1): 160–173, 2020.

[40] Zifu Wan, Pingping Zhang, Yuhao Wang, Silong Yong, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Sigma: Siamese mamba network for multi-modal semantic segmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1734–1744. IEEE, 2025.

[41] Chiao-Yi Wang, Faranguisse Kakhi Sadrieh, Yi-Ting Shen, Shih-En Chen, Sarah Kim, Victoria Chen, Achyut Raghavendra, Dongyi Wang, Osamah Saeedi, and Yang Tao. Memo: dataset and methods for robust multimodal retinal image registration with large or small vessel density differences. *Biomedical Optics Express*, 15(5):3457–3479, 2024.

[42] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12186–12195, 2022.

[43] Lekang Wen, Jing Xiao, Liang Liao, Jiajun Chen, and Mi Wang. CHARM: Collaborative harmonization across arbitrary modalities for modality-agnostic semantic segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(13):10603–10611, 2026.

[44] Dirui Xie, Xiaofang Hu, Zihan Wei, Zhengqiqi Yang, Yanlian Jiang, and Yue Zhou. Learning structural priors via laplacian rwkv diffusion with light-effect dataset for nighttime visibility enhancement. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4590–4599, 2025.

[45] Dongqing Xie, Yonghuang Wu, Zisheng Ai, Jun Min, Zhencun Jiang, Shaojin Geng, and Lei Wang. CCSD: Cross-modal compositional self-distillation for robust brain tumor segmentation with missing modalities. *Knowledge-Based Systems*, 346:116136, 2026.

[46] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16111–16121, 2024.

[47] Huangbiao Xu, Huanqi Wu, Xiao Ke, Junyi Wu, Rui Xu, and Jinglin Xu. Mcmoe: Completing missing modalities with mixture of experts for incomplete multimodal action quality assessment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(13):11241–11249, 2026.

[48] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J. Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25–28, 2024*. BMVA, 2024. Paper 755.

[49] Jun Yang, Lizhi Bai, Yaoru Sun, Chunqi Tian, Maoyu Mao, and Guorun Wang. Pixel difference convolutional network for rgb-d semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(3):1481–1492, 2024.

[50] Zhiwen Yang, Jiayin Li, Hui Zhang, Dan Zhao, Bingzheng Wei, and Yan Xu. Restore-rwkv: Efficient and effective medical image restoration with rwkv. *IEEE Journal of Biomedical and Health Informatics*, 30(1):513–526, 2026.

[51] Bo-Wen Yin, Jiao-Long Cao, Ming-Ming Cheng, and Qibin Hou. Dformerv2: Geometry self-attention for rgbd semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19345–19355, 2025.

[52] Hang Zhang, Allen A Zhang, Zishuo Dong, Anzheng He, Yang Liu, You Zhan, and Kelvin CP Wang. Robust semantic segmentation for automatic crack detection within pavement images using multi-mixing of global context and local image features. *IEEE Transactions on Intelligent Transportation Systems*, 25(9):11282–11303, 2024.

[53] Jiaming Zhang, Huayao Liu, Kailun Yang, Xinxin Hu, Ruiping Liu, and Rainer Stiefelhagen. Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. *IEEE Transactions on intelligent transportation systems*, 24(12):14679–14694, 2023.

[54] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1136–1147, 2023.

[55] Man Zhou, Xuanhua He, Danfeng Hong, and Bo Huang. Wkv-sharing embraced random shuffle rwkv high-order modeling for pan-sharpening. In *Advances in Neural Information Processing Systems*, volume 38, pages 158121–158142, 2025.

[56] Wujie Zhou, Enquan Yang, Jingsheng Lei, Jian Wan, and Lu Yu. Pgdenet: Progressive guided fusion and depth enhancement network for rgb-d indoor scene parsing. *IEEE Transactions on Multimedia*, 25:3483–3494, 2023.

[57] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62429–62442. PMLR, 2024.

[58] Qinfeng Zhu, Yuan Fang, Yuanzhi Cai, Cheng Chen, and Lei Fan. Rethinking scanning strategies with vision mamba in semantic segmentation of remote sensing imagery: An experimental study. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17: 18223–18234, 2024.

[59] Qiwei Zhu, Kai Li, Guojing Zhang, Xiaoying Wang, Jianqiang Huang, and Xilai Li. Gdsr: Global-detail integration through dual-branch network with wavelet losses for remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–16, 2025.

[60] Abid Hasan Zim, Aquib Iqbal, Zaid Al-Huda, Asad Malik, and Minoru Kuribayashi. Efficientcracknet: A lightweight model for crack segmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6279–6289. IEEE, 2025.

[61] Xin Zuo, Yu Sheng, Jifeng Shen, and Yongwei Shan. Topology-aware mamba for crack segmentation in structures. *Automation in Construction*, 168:105845, 2024.

A Theoretical Justification of Implicit Structural Alignment

In this section, we provide theoretical justifications for the key mechanisms in Prism. Rather than claiming general convergence guarantees, we focus on explaining how the proposed objectives and modules bias optimization and representation learning toward structurally meaningful regions. Specifically, we analyze focal gradient reweighting in PaKD, topology-aware serialization in PPI, and input-dependent sparse modulation in PulseMixer.

A.1 Proposition 1: Focal Gradient Reweighting for Structural Alignment

Proposition. *The focal heatmap alignment loss $\mathcal{L}_{hm}$ induces an adaptive gradient reweighting effect. Pixels with low teacher-student heatmap consistency receive larger effective gradient gains than high-consistency pixels. Under a separation assumption between skeleton and background regions, the expected gradient gain on skeleton pixels is substantially larger than that on background pixels, thereby biasing optimization toward structural regions during implicit alignment.*

Derivation. Let $u \in \Omega$ index a pixel location. We define the local teacher-student heatmap consistency as:

$$\hat{p}_u = \mathrm{clip}(1 - |\mathcal{H}_S(u) - \mathcal{H}_T(u)|\,,\ \epsilon,\ 1)\,, \quad \epsilon \in (0,1), \tag{22}$$

where $\mathcal{H}_S(u)$ and $\mathcal{H}_T(u)$ denote the student and teacher heatmap values at pixel u, respectively. A larger $\hat{p}_u$ indicates stronger consistency, while a smaller $\hat{p}_u$ corresponds to a harder pixel with larger structural discrepancy.

The pixel-wise focal heatmap alignment loss is defined as:

$$\mathcal{L}_{hm}^{(u)}(\hat{p}_u) = -(1-\hat{p}_u)^{\gamma}\log(\hat{p}_u), \quad \gamma > 0, \tag{23}$$

When $\gamma = 0$, this objective degenerates to the standard cross-entropy form $-\log(\hat{p}_u)$. When $\gamma > 0$, the factor $(1-\hat{p}_u)^{\gamma}$ suppresses already-aligned pixels and emphasizes pixels with low teacher-student consistency.

For pixels whose consistency scores are not saturated by the clipping operation, differentiating $\mathcal{L}_{hm}^{(u)}$ with respect to $\hat{p}_u$ gives:

$$\frac{\partial \mathcal{L}_{hm}^{(u)}}{\partial \hat{p}_u} = \gamma(1-\hat{p}_u)^{\gamma-1}\log(\hat{p}_u) - \frac{(1-\hat{p}_u)^{\gamma}}{\hat{p}_u}, \tag{24}$$

Applying the chain rule to the student parameters θ yields:

$$\nabla_\theta \mathcal{L}_{hm} = \sum_{u\in\Omega} \frac{\partial \mathcal{L}_{hm}^{(u)}}{\partial \hat{p}_u} \nabla_\theta \hat{p}_u, \tag{25}$$

Taking norms gives the following upper bound:

$$\|\nabla_\theta \mathcal{L}_{hm}\| \le \sum_{u\in\Omega} \beta(\hat{p}_u)\|\nabla_\theta \hat{p}_u\|, \quad \beta(\hat{p}) \triangleq \left|\gamma(1-\hat{p})^{\gamma-1}\log(\hat{p}) - \frac{(1-\hat{p})^{\gamma}}{\hat{p}}\right|, \tag{26}$$

Here, $\beta(\hat{p}_u)$ acts as a dynamic per-pixel gradient gain.

For low consistency $\hat{p} \to 0^+$, the second term dominates and gives:

$$\beta(\hat{p}) = \Theta\left(\frac{1}{\hat{p}}\right), \tag{27}$$

indicating that highly inconsistent pixels receive large gradient gains.

Conversely, for high consistency $\hat{p} \to 1^-$, using $\log(\hat{p}) \approx -(1-\hat{p})$ gives:

$$\beta(\hat{p}) = \Theta((1-\hat{p})^\gamma), \tag{28}$$

which approaches zero as $\hat{p}$ approaches one. Therefore, the focal heatmap loss naturally reduces the contribution of easy, already-aligned pixels and allocates larger gradient gains to hard pixels.

We next relate this reweighting effect to structural alignment. Let Ω_{skel} denote skeleton or boundary pixels and Ω_{bg} denote background pixels. We assume a mild separation condition:

$$\hat{p}_u \le \delta,\ \forall u \in \Omega_{skel}, \qquad \hat{p}_u \ge 1-\eta,\ \forall u \in \Omega_{bg}, \tag{29}$$

where $\delta, \eta \in (0,1)$, with small δ indicating low skeleton consistency and small η indicating high background consistency.

We also assume the intrinsic sensitivities are bounded:

$$0 < c_g \le \|\nabla_\theta \hat{p}_u\| \le C_g < \infty, \tag{30}$$

Under these assumptions, the expected effective gradient contribution on skeleton and background regions satisfies the order-level relation:

$$\frac{\mathbb{E}_{u\in\Omega_{skel}}[\beta(\hat{p}_u)\|\nabla_\theta \hat{p}_u\|]}{\mathbb{E}_{u\in\Omega_{bg}}[\beta(\hat{p}_u)\|\nabla_\theta \hat{p}_u\|]} \gtrsim \frac{c_g}{C_g}\cdot\frac{1/\delta}{\eta^\gamma}, \tag{31}$$

Thus, when skeleton pixels are harder to align and background pixels are already consistent, the optimization process assigns much larger effective gradient weight to structural regions.

Conclusion. This proposition shows that $\mathcal{L}_{hm}$ does not uniformly penalize all pixels. Instead, it introduces a dynamic gain $\beta(\hat{p}_u)$ that emphasizes low-consistency regions and suppresses high-consistency regions. Under the stated separation and bounded-sensitivity assumptions, skeleton and boundary pixels receive substantially larger effective gradient contributions than background pixels, providing a mechanism-level explanation for why PaKD encourages implicit structural alignment.

B Theoretical Justification of Sparse Gated Pulse Propagation

In this section, we analyze the mechanisms behind Pulse Propagation Insight (PPI) and Sparse Gating. The analysis explains why PPI can reduce topology distortion under ordering regularity assumptions, and why sparse gating increases the effective diversity of linear transformations through input-dependent modulation.

B.1 Proposition 2: Distortion Reduction under PPI Ordering Regularity

Proposition. *Consider crack images whose foreground contains thin, approximately one-dimensional structures embedded in a 2D grid. Under assumptions of inter-component separation and locally regular PPI ordering, PPI admits a bounded expected sequential gap along crack-manifold edges and controls false proximity across disconnected regions. Consequently, PPI yields a smaller structure-aware distortion bound than rigid raster ordering or foreground-first ordering in the multi-component crack regime.*

Derivation. We model an image as an 8-neighbor grid graph $\mathcal{G} = (\Omega, \mathcal{E})$, where Ω is the pixel set and $\mathcal{E}$ is the set of undirected 8-neighbor edges. Let $\pi : \Omega \to \{1, \dots, |\Omega|\}$ be a serialization function. We define $\mathcal{E}_{crk} \subseteq \mathcal{E}$ as the set of edges located on or near the crack manifold. To measure topology mismatch induced by a 2D-to-1D ordering, we consider the following structure-aware distortion energy:

$$\mathcal{J}(\pi) = \sum_{(u,v)\in\mathcal{E}_{crk}} (1 - \exp(-w|\pi(u)-\pi(v)|)) + \sum_{\substack{(u,v)\notin\mathcal{E} \\ |\pi(u)-\pi(v)|\le\tau}} \Theta(\|\mathbf{p}_u - \mathbf{p}_v\|_2), \tag{32}$$

where $w > 0$ controls the penalty strength for large sequential gaps, τ is a small sequence-window threshold, $\mathbf{p}_u \in \mathbb{R}^2$ is the 2D coordinate of pixel u, and $\Theta(\cdot)$ is a nonnegative nondecreasing distance penalty. The first term penalizes breaking local crack-manifold continuity, while the second term penalizes false proximity, where spatially distant non-neighboring pixels become close in the 1D sequence.

We use the following assumptions. First, different crack components are separated by at least $d_0 > 0$, and the image has finite diameter $D_{\max}$, such that $\|\mathbf{p}_u - \mathbf{p}_v\|_2 \le D_{\max}$ for all $u, v \in \Omega$. Hence, for false proximity pairs:

$$\Theta(d_0) \le \Theta(\|\mathbf{p}_u - \mathbf{p}_v\|_2) \le \Theta(D_{\max}), \tag{33}$$

Second, PPI sorts pixels according to a pulse distance field with a fixed component-consistent tie-breaking rule. We assume the resulting ordering is locally regular on the crack manifold, namely:

$$\mathbb{E}_{(u,v)\in\mathcal{E}_{crk}}[|\pi_{\mathrm{ppi}}(u) - \pi_{\mathrm{ppi}}(v)|] \le C_{\mathrm{ppi}}, \tag{34}$$

where C_{ppi} is independent of the image width W. This assumption formalizes the intended behavior of PPI: adjacent pixels along the crack manifold are assigned nearby sequence positions in expectation.

For raster scanning, vertically adjacent pixels have a sequence gap proportional to the image width:

$$|\pi_{\mathrm{ras}}(u) - \pi_{\mathrm{ras}}(v)| = W, \tag{35}$$

Therefore, for vertical crack-manifold edges $\mathcal{E}_{vert} \subseteq \mathcal{E}_{crk}$:

$$\sum_{(u,v)\in\mathcal{E}_{vert}} \left(1 - \exp\left(-w|\pi_{\mathrm{ras}}(u) - \pi_{\mathrm{ras}}(v)|\right)\right) = |\mathcal{E}_{vert}|\left(1 - \exp(-wW)\right), \tag{36}$$

When W is large, raster scanning incurs a large continuity penalty for vertical or oblique crack structures.

For foreground-first ordering, pixels from disconnected foreground components may become adjacent in the sequence. Let N^{π}_{false} denote the number of false proximity events:

$$N^{\pi}_{\mathrm{false}} = \left|\{(u,v) : (u,v) \notin \mathcal{E},\ |\pi(u) - \pi(v)| \le \tau\}\right|, \tag{37}$$

If false proximity occurs across components separated by at least d_0, then the penalty is at least $\Theta(d_0)$ per event.

In contrast, under component-consistent tie-breaking, PPI avoids many abrupt transitions between disconnected crack components. We express this condition as:

$$N^{\pi_{\mathrm{ppi}}}_{\mathrm{false}} \le N^{\pi_{\mathrm{fg}}}_{\mathrm{false}}, \tag{38}$$

where π_{fg} denotes a foreground-first ordering.

Combining the local regularity assumption in Eq. (34) with the bounded image diameter gives the following upper bound for PPI:

$$\mathbb{E}[\mathcal{J}(\pi_{\mathrm{ppi}})] \le |\mathcal{E}_{crk}|\left(1 - \exp(-wC_{\mathrm{ppi}})\right) + N^{\pi_{\mathrm{ppi}}}_{\mathrm{false}}\Theta(D_{\max}), \tag{39}$$

The first term is independent of image width W, while raster scanning contains a width-dependent continuity penalty. The second term is controlled by the number of false proximity events, which is reduced when PPI uses component-consistent ordering.

Conclusion. This proposition does not claim that PPI is universally optimal for all possible images. Instead, it shows that under local ordering regularity and component separation, PPI admits a favorable structure-aware distortion bound. By keeping neighboring crack-manifold pixels close in the sequence and reducing harmful false proximity across disconnected components, PPI provides a theoretical explanation for why pulse-based serialization better preserves curvilinear crack topology than rigid raster or foreground-first ordering.

B.2 Proposition 3: Input-Dependent Sparse Gating as a Union-of-Subspaces Model

Proposition. *Sparse gating transforms a static linear recurrence into an input-dependent piecewise linear mapping. Over data that activates multiple gating patterns, the reachable representations form a union of subspaces. When the activated subspaces contain complementary directions, their global*

span can exceed the span induced by any single static or single-gated operator, thereby alleviating the effective static parameterization bottleneck.

Derivation. For a static linear operator $\mathbf{W} \in \mathbb{R}^{d\times d}$, the reachable representation space is contained in the image of $\mathbf{W}$:

$$\mathcal{V}_{\text{static}} = \{\mathbf{W}\mathbf{x} \mid \mathbf{x} \in \mathcal{X}\} \subseteq \text{Im}(\mathbf{W}), \qquad \dim(\text{Im}(\mathbf{W})) = \text{rank}(\mathbf{W}), \tag{40}$$

where $\mathcal{X}$ denotes the data support. This static operator applies the same transformation to all inputs, which may be insufficient for complex crack manifolds with varying topology, background noise, and missing-modality patterns.

In Prism, sparse gating modulates the WKV output with an input-dependent gate. For theoretical clarity, we analyze a hard-thresholded version of the sparse gate, which can be viewed as a discrete approximation to the continuous thresholded gate used in the actual module. The simplified gated mapping is:

$$\mathbf{y} = \mathbf{D}(\mathbf{x})\mathbf{W}\mathbf{x}, \qquad \mathbf{D}(\mathbf{x}) = \text{diag}(\mathbb{I}(\mathbf{W}_g\mathbf{x} + \mathbf{b}_g > 0)), \tag{41}$$

where $\mathbf{D}(\mathbf{x})$ is an input-dependent diagonal mask, $\mathbf{W}_g$ and $\mathbf{b}_g$ are gating parameters, and $\mathbb{I}(\cdot)$ is the indicator function.

Because the gate is piecewise constant under a hard threshold, the input space is partitioned into regions $\{P_k\}_{k\in\mathcal{K}}$ such that:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}_k, \quad \forall \mathbf{x} \in P_k, \tag{42}$$

Within each region, the dynamic mapping becomes:

$$\mathcal{T}_{\text{dyn}}(\mathbf{x}) = \mathbf{D}_k\mathbf{W}\mathbf{x}, \tag{43}$$

Thus, over all activated regions, the reachable representation set is contained in a union of subspaces:

$$\mathcal{V}_{\text{dyn}} \subseteq \bigcup_{k\in\mathcal{K}} \text{Im}(\mathbf{D}_k\mathbf{W}), \tag{44}$$

The dimension of the global span satisfies:

$$\max_{k\in\mathcal{K}} \text{rank}(\mathbf{D}_k\mathbf{W}) \le \dim(\text{span}(\mathcal{V}_{\text{dyn}})) \le \min\left\{d, \sum_{k\in\mathcal{K}} \text{rank}(\mathbf{D}_k\mathbf{W})\right\}, \tag{45}$$

This bound is more appropriate than assuming the dynamic span always exceeds $\text{rank}(\mathbf{W})$, since a particular gate may suppress dimensions. The key advantage appears at the dataset level: different samples can activate different masks, and the union of their induced subspaces can cover complementary directions.

If there exist two activated regions $k \neq k'$ such that:

$$\text{Im}(\mathbf{D}_k\mathbf{W}) \not\subseteq \text{Im}(\mathbf{D}_{k'}\mathbf{W}), \qquad \text{Im}(\mathbf{D}_{k'}\mathbf{W}) \not\subseteq \text{Im}(\mathbf{D}_k\mathbf{W}), \tag{46}$$

then the global span strictly exceeds the span of at least one single gated operator:

$$\dim(\text{span}\left(\text{Im}(\mathbf{D}_k\mathbf{W}) \cup \text{Im}(\mathbf{D}_{k'}\mathbf{W})\right)) > \max\left\{\text{rank}(\mathbf{D}_k\mathbf{W}), \text{rank}(\mathbf{D}_{k'}\mathbf{W})\right\}, \tag{47}$$

When the original static recurrence is effectively rank-constrained or dominated by a single low-dimensional response pattern, this union-of-subspaces behavior provides a mechanism for increasing effective representational diversity.

The continuous sparse gate used in Prism can be interpreted as a soft relaxation of the hard mask above:

$$\mathbf{g}(\mathbf{x}) = [\sigma(\mathbf{W}_g\mathbf{x} + \mathbf{b}_g) - \theta]_+, \tag{48}$$

Instead of selecting binary subspaces, it continuously rescales channels according to input-dependent confidence. Therefore, the actual module preserves the same principle while allowing smoother optimization.

Conclusion. Sparse gating converts a single static transformation into an input-dependent family of transformations. Across a dataset, different samples can activate different channel subsets and therefore induce a union of complementary subspaces. This provides a theoretical explanation for how sparse gating alleviates the static parameterization bottleneck of linear recurrence and improves the capacity to model complex crack manifolds under missing-modality conditions.

C More Analysis of UCPF

The UCPF module is designed to address two fundamental limitations in multimodal crack segmentation: the spatial inconsistency of modal reliability and the topological fragmentation caused by sensor noise. Unlike standard channel-wise concatenation or global attention mechanisms which assume uniform reliability distribution, UCPF introduces a pixel-wise competition mechanism. Specifically, we employ a lightweight estimator to predict the uncertainty map for each modality. By inverting this uncertainty, we obtain a normalized confidence weight $\mathbf{w}_m$ for the m-th modality:

$$\mathbf{w}_m = \frac{1 - \sigma(\mathcal{H}_{unc}(\mathbf{F}_m))}{\sum_{k=1}^{M} (1 - \sigma(\mathcal{H}_{unc}(\mathbf{F}_k))) + \epsilon}, \tag{49}$$

where $\mathcal{H}_{unc}$ serves as a pixel-level arbiter. The term $1-\sigma(\cdot)$ effectively converts high uncertainty into low blending weights. This ensures that the fused representation $\mathbf{F}_{fused}$ is dynamically dominated by the most reliable sensor at each specific coordinate, creating a robust feature base that filters out local interference such as shadows or thermal blur.

Building upon this reliable fusion, UCPF addresses the challenge of topological disconnection through a confidence-gated propagation strategy. Standard convolutions operate isotropically, often blurring boundaries or failing to bridge gaps. To solve this, we explicitly model the feature flow as a diffusion process originating from high-confidence regions. We define the propagation flow $\mathbf{F}_{flow}$ as:

$$\mathbf{F}_{flow} = \mathcal{K}_{prop} \left(\mathbf{F}_{fused} \odot \sigma(\mathcal{H}_{conf}(\mathbf{F}_{fused})) \right), \tag{50}$$

where the pre-gating operation $\mathbf{F}_{fused} \odot \sigma(\mathcal{H}_{conf})$ acts as a semantic filter, ensuring that only features with high structural probability are selected as "seed points." The large-kernel grouped convolution $\mathcal{K}_{prop}$ then diffuses these seed features into the neighborhood. This mechanism functions as semantic inpainting, where information flows from clear crack segments to low-confidence breakpoints, effectively healing topological gaps without hallucinating features in the background.

Finally, to incorporate this topological healing while mitigating the over-smoothing often associated with large-kernel operations, we employ a conditional residual injection mechanism. Instead of adding the propagated features indiscriminately, we utilize a gated formulation for the final reconstruction:

$$\mathbf{F}_{rec} = \text{BN} \left(\mathbf{F}_{fused} + \sigma(\mathcal{H}_{gate}(\mathbf{F}_{flow})) \odot \mathbf{F}_{flow} \right), \tag{51}$$

In this formulation, $\mathcal{H}_{gate}$ generates a spatial attention map that determines where the "healing" is required. The gate tends to open in regions with structural ambiguity to inject contextual information, and close in clean background regions to preserve original details. This design strikes a critical balance, allowing UCPF to aggressively restore connectivity in fracture zones while maintaining high precision in non-defect areas.

D Additional Experimental Details

In this section, we provide detailed specifications regarding the teacher model configuration and the evaluation protocols used to validate Prism. For distillation, we specifically utilize the **sam2_hiera_base_plus** checkpoint as the frozen teacher network. We prioritize SAM2 over the original SAM or other static foundation models for two strategic reasons: First, unlike the heavy plain backbones used in previous iterations, SAM2 adopts the Hiera architecture, which allows for more effective multi-scale feature extraction essential for detecting fine cracks. Second, and more importantly, SAM2 is architecturally optimized for video object segmentation with inherent temporal coherence mechanisms; this design enables its feature manifold to better capture the continuous topological nature of cracks rather than treating them as disjoint static patches, providing a more geometrically consistent supervision signal for our pulse propagation. It is crucial to note that the SAM2 teacher is frozen during the entire training phase and is strictly excluded from the inference process. Consequently, the heavy computational burden of the teacher model does not impact the real-time deployment of Prism, ensuring zero overhead during real-world application.

To comprehensively evaluate the performance of Prism, we adopted the **Binary F1 score** and mean Intersection over Union (mIoU) metrics. The Binary F1 is utilized to quantify the harmonic mean of precision and sensitivity **specifically for the positive crack class**, defined as:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \tag{52}$$

where TP, FP, and FN denote the pixel counts for true positives, false positives, and false negatives of the crack category, respectively. Complementarily, to assess region-level consistency, mIoU calculates the average overlap ratio between the predicted segmentation mask and the ground truth across all classes **(including background)**:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{53}$$

where k represents the number of foreground classes (here $k = 1$), and i indicates the class index (with $i = 0$ typically denoting background). **It is worth noting that since mIoU incorporates the typically high-accuracy background class into the average, while the Binary F1 focuses exclusively on the minority crack class, the resulting mIoU values may numerically exceed the F1 scores due to class imbalance.** These metrics are critical for assessing boundary precision and topological integrity in this task [24].

Given the practical constraints of edge deployment, we also strictly monitored the computational complexity. The number of Parameters (Params), Floating Point Operations (FLOPs), and physical Model Size were utilized to measure the efficiency of the models [25], verifying that Prism maintains a lightweight architecture suitable for drone-based environments.

Furthermore, regarding the simulation of missing modalities, we diverge from the standard paradigm that assumes the complete unavailability of a specific sensor. Instead, we implement a flexible Stochastic Sample-Level Dropout protocol to mirror complex industrial realities. We simulate scenarios where data loss occurs stochastically at arbitrary ratios within any modality; for instances of missing data, zero-padding is applied to the respective input channels to represent signal loss. This granular approach offers a strategic advantage over binary modality dropout: it accurately models sporadic sensor instability, such as temporary transmission jitter, glare-induced blinding, or frame drops common in drone operations. By training under these non-uniform conditions, Prism is forced to develop robust feature reconstruction capabilities that adapt to unpredictable, instance-level data unavailability rather than merely learning to rely on a fixed subset of modalities.

E Diagnostics for Structural Learning

E.1 Serialization Continuity of PPI

To directly examine whether PPI preserves crack-neighborhood continuity during serialization, we evaluate six ordering strategies on the same ground-truth-defined CrackPolar skeleton manifold, so prediction quality does not affect their comparison. The ground truth is used only for this offline diagnostic; the actual PPI module derives its ordering from student scores and does not access the ground truth. For each 8-neighbor skeleton edge, the serialization gap is the absolute difference between the sequence positions of its endpoints. R-edge@8 denotes the percentage of these edges with a gap no greater than 8, and J-cont is the mean of $1 - \exp(-0.01\,\mathrm{gap})$.

Table 8: Serialization continuity on the CrackPolar skeleton manifold. Blue bold and green underlined entries denote best and second-best results.

Ordering	R-edge@8 (%)↑	Mean Gap↓	Gap P95↓	J-cont↓
Raster	17.39	105.9	128.0	0.5981
Q-Shift	17.59	105.9	181.1	0.5761
SAVSS	74.64	104.2	895.1	0.1765
Hilbert	71.29	129.7	268.5	0.1379
Foreground-first	<u>85.59</u>	<u>4.628</u>	<u>12.1</u>	<u>0.0436</u>
PPI	**96.93**	**2.268**	**4.774**	**0.0220**

As shown in Table 8, PPI preserves 96.93% of neighboring skeleton edges within a sequence gap of 8 and reduces J-cont by 49.5% relative to foreground-first ordering. The lower mean and tail gaps further indicate that PPI serializes connected crack pixels more continuously rather than merely placing foreground pixels before the background.

E.2 Gradient Concentration of PaKD

We further measure how PaKD redistributes learning signals across crack structures without retraining. The settings $\gamma = 0$ and $\gamma = 2$ act on identical heatmaps; ground truth is used only for the existing prompt target and the offline diagnostic partition into skeleton, boundary, and background. For each image, we compute the mean absolute output gradient in these regions and report the median skeleton-to-background and boundary-to-background ratios. The standard objective corresponds to $\gamma = 0$, while PaKD uses the focal setting $\gamma = 2$.

Table 9: Gradient concentration at different training stages. Blue bold and green underlined entries denote the best and second-best results within each stage.

Objective	Stage	Skeleton/BG↑	Boundary/BG↑
Standard ($\gamma = 0$)	Early (epoch 30)	<u>11.843</u>	<u>3.381</u>
PaKD focal ($\gamma = 2$)	Early (epoch 30)	**59.882**	**8.667**
Standard ($\gamma = 0$)	Middle (epoch 60)	<u>34.081</u>	<u>6.442</u>
PaKD focal ($\gamma = 2$)	Middle (epoch 60)	**315.737**	**49.809**
Standard ($\gamma = 0$)	Late (epoch 90)	<u>12.973</u>	<u>2.879</u>
PaKD focal ($\gamma = 2$)	Late (epoch 90)	**87.652**	**13.862**

Table 9 shows that PaKD increases skeleton-gradient concentration by 5.06×, 9.26×, and 6.76× at the early, middle, and late stages, respectively. The corresponding boundary-gradient gains are 2.56×, 7.73×, and 4.82×. This verifies that the focal prompt-distillation objective consistently concentrates supervision on the sparse structural regions rather than the dominant background.

F Details of Objective Function

As elaborated in section 3.5, the overall objective function of Prism, denoted as $\mathcal{L}_{total}$, comprises four components: the segmentation loss $\mathcal{L}_{seg}$, the probability map distillation loss $\mathcal{L}_{prob}$, the PaKD distillation loss $\mathcal{L}_{pakd}$, and the PPI skeleton loss $\mathcal{L}_{ppi}$. While the detailed calculation procedures for $\mathcal{L}_{prob}$, $\mathcal{L}_{pakd}$, and $\mathcal{L}_{ppi}$ have been presented in the Methodology section of the main text, herein we provide a supplementary detailed definition and formulation of the hybrid segmentation loss $\mathcal{L}_{seg}$, which supervises the final segmentation output.

$\mathcal{L}_{seg}$ is designed to ensure high consistency between the model-generated crack segmentation map and the ground truth, both in terms of pixel-level classification accuracy and overall structural overlap. To mitigate the extreme class imbalance inherent in crack detection tasks, where crack pixels are significantly scarcer than background pixels, we employ a hybrid strategy combining Binary Cross-Entropy Loss ($\mathcal{L}_{BCE}$) and Dice Loss ($\mathcal{L}_{Dice}$). Specifically, let $\mathbf{Y} \in \{0,1\}^{H\times W}$ denote the ground truth binary mask and $\mathbf{P} \in [0,1]^{H\times W}$ denote the predicted probability map. We define $\mathcal{L}_{seg}$ as:

$$\mathcal{L}_{seg} = \mathcal{L}_{BCE}(\mathbf{P}, \mathbf{Y}) + \mathcal{L}_{Dice}(\mathbf{P}, \mathbf{Y}), \tag{54}$$

The BCE loss function primarily to quantify pixel-level classification accuracy. It independently evaluates the divergence between the predicted probability distribution and the true distribution for each pixel, effectively compelling the model to classify each pixel correctly. Its mathematical expression is simplified as:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log p_i + (1-y_i)\log(1-p_i)\right], \tag{55}$$

where N denotes the total number of pixels in the image, y_i represents the ground truth label of the i-th pixel, and p_i indicates the predicted probability of the i-th pixel being a crack. $\mathcal{L}_{BCE}$ exhibits stability when processing the dominant background pixels, providing smooth gradient descent directions during optimization.

Derived from the Dice coefficient, the Dice loss serves as a metric based on set overlap. Unlike $\mathcal{L}_{BCE}$, $\mathcal{L}_{Dice}$ emphasizes the overlap ratio between the predicted and ground truth regions, thereby possessing inherent robustness to class imbalance. In the context of crack segmentation, it effectively prevents the model from converging to local optima dominated by background pixels. The calculation formula is defined as:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N} p_i y_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} y_i + \epsilon}, \tag{56}$$

where the numerator approximates the intersection between the predicted and ground truth regions, and the denominator represents their union. The term ϵ is a small smoothing constant introduced to prevent division by zero and stabilize the training process.

By jointly optimizing $\mathcal{L}_{BCE}$ and $\mathcal{L}_{Dice}$, Prism balances micro-level pixel classification accuracy with macro-level topological structural integrity. This enables the generation of segmentation results with sharp boundaries and strong connectivity, even in scenarios involving missing multimodal data.

G Details of GPM

The GPM is designed to overcome the limitations of standard convolutions in processing slender crack structures by jointly modeling spatial anisotropy, frequency-domain connectivity, and topological diffusion characteristics.

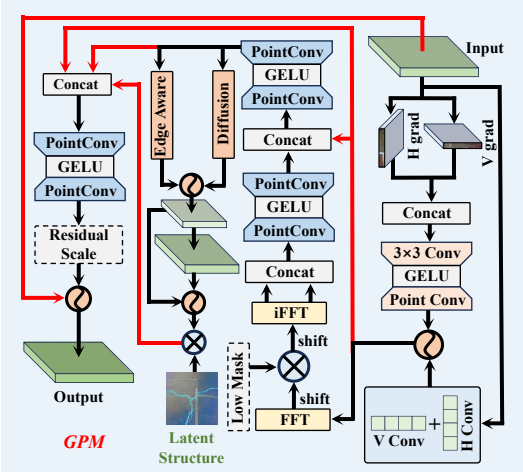


Figure 5: The structure of GPM. It enhances the model's perception ability for elongated cracks by modeling anisotropy, frequency domain connectivity, and topological diffusion.

As shown in Figure 5, to address the significant directional sensitivity of cracks, we employ the Sobel operator to calculate the gradient magnitudes of the input feature $\mathbf{X}$ in both horizontal and vertical directions. Based on these gradients, we generate direction-adaptive strip features $\mathbf{F}_{strip}$ as follows:

$$\mathbf{F}_{strip} = \sigma(\Phi_{gate}(|\nabla_x \mathbf{X}|)) \odot \mathcal{K}_{1\times L}(\mathbf{X}) + \sigma(\Phi_{gate}(|\nabla_y \mathbf{X}|)) \odot \mathcal{K}_{L\times 1}(\mathbf{X}), \tag{57}$$

where $|\nabla_x \mathbf{X}|$ and $|\nabla_y \mathbf{X}|$ denote the gradient magnitude maps of the input feature $\mathbf{X}$ in the horizontal and vertical directions, respectively, capturing the edge orientation of the cracks, Φ_{gate} is a lightweight convolutional projection layer used to map the gradient information into directional weights, σ represents the Sigmoid activation function. $\mathcal{K}_{1\times L}$ and $\mathcal{K}_{L\times 1}$ denote large-receptive-field strip convolutions with kernel sizes of $1 \times L$ and $L \times 1$, responsible for capturing long-range dependencies in the horizontal and vertical directions, respectively.

Simultaneously, to enhance crack connectivity from a global perspective, separated frequency components are weighted and fused in the frequency domain:

$$\mathbf{F}_{freq} = \Phi_{freq}\Big(\mathrm{Cat}\Big(\mathcal{F}^{-1}(\mathbf{Y}_{low} \odot \mathbf{W}_{low}),\ \mathcal{F}^{-1}(\mathbf{Y}_{high} \odot \mathbf{W}_{high})\Big)\Big), \tag{58}$$

where $\mathcal{F}^{-1}$ denotes the 2D Inverse Fourier Transform, $\mathbf{Y}_{low}$ and $\mathbf{Y}_{high}$ represent the low-frequency components and high-frequency components separated by a central mask, $\mathbf{W}_{low}$ and $\mathbf{W}_{high}$ are frequency-domain filter weights automatically learned during training to dynamically enhance specific frequency responses, Φ_{freq} is a non-linear convolutional layer used for channel fusion.

Topology-Aware Diffusion simulates the physical propagation process of cracks. Letting the initial state $\mathbf{H}_0$ be the concatenation of spatial and frequency features, the feature update at step t follows the dynamic equation:

$$\mathbf{H}_t = \sigma(\Phi_{diff}(\mathbf{H}_{t-1})) \odot \mathcal{K}_{diff}(\mathbf{H}_{t-1}) + (1 - \sigma(\Phi_{diff}(\mathbf{H}_{t-1}))) \odot \mathcal{K}_{edge}(\mathbf{H}_{t-1}), \quad (59)$$

where $\mathcal{K}_{diff}$ is a large-kernel diffusion convolution operator used to simulate crack propagation along the topological structure, $\mathcal{K}_{edge}$ is an edge-preserving convolution designed to retain high-frequency boundary information, Φ_{diff} is a projection layer that generates a dynamic diffusion gate, the term $\sigma(\Phi_{diff}(\mathbf{H}_{t-1}))$ determines whether the current pixel tends towards feature diffusion or edge preservation, thereby repairing discontinuities while preventing over-smoothing.

To adapt to drastic variations in crack width, GPM computes features with different receptive fields in parallel and performs adaptive weighting based on the local feature variance $\text{Var}(\mathbf{H}_T)$:

$$\mathbf{F}_{scale} = \mathbf{S}_{sma} \odot \mathcal{K}_{sma}(\mathbf{H}_T) + \mathbf{S}_{lar} \odot \mathcal{K}_{lar}(\mathbf{H}_T), \quad (60)$$

where $\mathbf{S}_{sma}$ and $\mathbf{S}_{lar}$ are normalized scale weights obtained by applying the Softmax function to the local variance features $\Phi_{scale}(\text{Var}(\mathbf{H}_T))$. $\mathcal{K}_{sma}$ and $\mathcal{K}_{lar}$ represent small-kernel convolutions for capturing fine cracks and large-kernel convolutions for capturing coarse cracks.

Finally, the output $\mathbf{F}_{out}$ integrates features from all branches via a residual connection:

$$\mathbf{F}_{out} = \mathbf{X} + \Psi_{proj}(\text{Cat}(\mathbf{F}_{strip}, \mathbf{F}_{freq}, \mathbf{F}_{scale})), \quad (61)$$

where Ψ_{proj} denotes the final feature projection layer, which maps the multi-branch features back to the original channel dimension for residual addition with the input $\mathbf{X}$, ensuring effective gradient propagation.

H Additional Visualization Analysis

In this section, we provide a comprehensive visual analysis to empirically validate the theoretical propositions. Specifically, we investigate how the student model Prism achieves implicit structural alignment with the teacher SAM2, and how the Pulse Propagation Insight mechanism evolves during training to progressively reduce manifold distortion.

To intuit the mechanism of the PaKD, we visualize the intermediate feature manifolds and probability responses. The visualization demonstrates two critical phases: Probabilistic Manifold Convergence and the Topological Completion.

We first analyze the probability maps generated by the frozen teacher SAM2 and the student Prism. As illustrated in Figure 6 and 7, the teacher network, guided by optimal point prompts, generates a highly confident and sharp probability distribution along the crack skeleton. Crucially, although Prism is a lightweight model without access to the heavy image encoder of SAM2 during inference, its predicted probability map, labeled as MoRe Prob, exhibits a striking morphological similarity to that of the teacher. The student not only captures the main structural ridge but also mimics the uncertainty gradients at the blurry boundaries. This visual evidence confirms Theorem A.1, demonstrating that the combination of the probability map distillation loss and the focal heatmap alignment loss successfully forces the latent representation of the student to converge to the optimal manifold defined by the teacher.

To understand how the model learns to minimize Manifold Distortion Energy as defined in Theorem B.1, we visualize the training evolution of the PPI module from the initial state to convergence. Figure 8 displays the Skeleton Map, the Geodesic Distance Map, and the resulting PPI Insight Order, where brighter colors indicate earlier positions in the sequence and darker colors indicate later positions.

At Epoch 0, the model exhibits a stochastic state. The Skeleton Map is noisy with high entropy, and consequently, the PPI Order appears chaotic and scattered. This corresponds to a high-distortion state where the scanning sequence lacks spatial coherence, resembling a random walk that fails to preserve local neighborhood correlations.

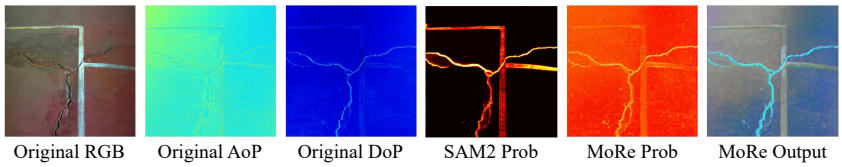


Figure 6: Visualization of the outputs from SAM2 and MoRe.

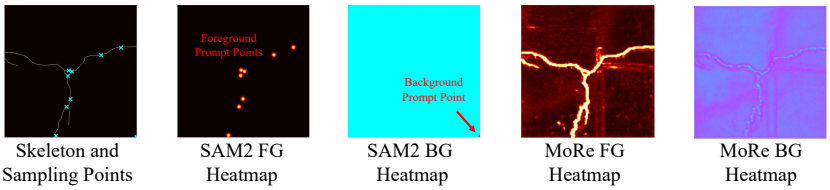


Figure 7: Visualization of the skeleton map, prompt points, and heatmap generated by PaKD.

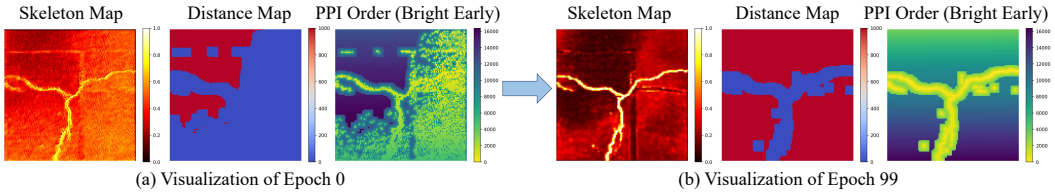


Figure 8: Visualization of PPI's skeleton feature, distance map, and insight order.

In stark contrast, at Epoch 99, a clear structural emergence is observed. The Skeleton Map accurately delineates the crack topology. Most importantly, the PPI Order map visualizes the theoretical concept of the pulse: the bright regions are strictly concentrated along the crack skeleton, while the colors gradient smoothly into darker shades towards the background. This gradient visualization serves as direct empirical proof of the Geodesic Continuity derived in Theorem B.1. It demonstrates that the model has learned to dynamically arrange the sequence such that spatially adjacent pixels on the intrinsic manifold are placed adjacently in the temporal sequence. By starting the scan at the crack and diffusing outward, Prism effectively eliminates the False Proximity effect common in raster scanning, ensuring that the recurrent features flow naturally from high-information foregrounds to the background context.

I More Comparative Experiments

To comprehensively evaluate the generalization capability of Prism under varying input conditions, we conducted additional experiments on the CrackPolar dataset using dual-modal input settings.

I.1 Experiments on CrackPolar Dual-modal Input

As presented in Table 10 and Figure 9, extensive testing across various missing rates demonstrates that Prism consistently achieves SOTA performance.

Specifically, within the RGB and DoP modality combination, even under the extreme scenario where 90% of the auxiliary DoP data is missing, Prism maintains an F1 of 0.7197, outperforming the second-best method, Sigma [40], by 5.13%. Furthermore, in the RGB and AoP combination, when the primary RGB modality suffers from a 70% missing rate, Prism sustains high performance with an F1 of 0.7075 and an mIoU of 0.7838. These results represent improvements of 4.83% and 1.82%, respectively, over MCubeS [19], which otherwise performs well in this setting. These findings indicate that the MoRe module, leveraging structured knowledge distilled via PaKD, effectively infers and completes the latent features of missing modalities based on available data. This capability significantly enhances the model's robustness in missing-modality scenarios, effectively mitigating performance degradation.

Table 10: Performance comparison under dual-modal and multi missing settings. **Purple** and Green denote best and second-best results.

	Missing Ratio		CMX		PrimKD		Sigma		CMNeXT		MCubeS		GminiFu		PWRF		LIDAR		**Prism**	
	RGB	DoP	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
CrackPolar	10%	Full	0.7197	0.7923	0.7233	0.7963	0.7242	0.7953	0.7043	0.7854	0.6977	0.7893	0.6714	0.7637	0.6719	0.7742	0.6935	0.7751	**0.7434**	**0.8005**
	30%	Full	0.7036	0.7870	0.7164	0.7841	0.6873	0.7746	0.6728	0.7677	0.6773	0.7746	0.6449	0.7487	0.6404	0.7546	0.6430	0.7410	**0.7269**	**0.7938**
	50%	Full	0.6866	0.7779	0.6957	**0.7831**	0.6793	0.7771	0.6635	0.7649	0.6687	0.7695	0.6332	0.7432	0.6128	0.7434	0.6345	0.7372	**0.7122**	0.7822
	70%	Full	0.6757	0.7655	0.6882	0.7660	0.6724	0.7691	0.6502	0.7524	0.6488	0.7585	0.6233	0.7407	0.6105	0.7355	0.6366	0.7388	**0.7094**	**0.7811**
	90%	Full	0.6738	0.7596	0.6791	0.7600	0.6703	0.7628	0.6414	0.7533	0.6295	0.7481	0.6018	0.7195	0.5960	0.7286	0.6266	0.7245	**0.7193**	**0.7892**
	Full	10%	0.7197	0.7923	0.7169	0.7958	0.7271	0.7970	0.7026	0.7886	0.6984	0.7902	0.6656	0.7587	0.6891	0.7744	0.7049	0.7818	**0.7446**	**0.8013**
	Full	30%	0.7160	0.7909	0.7106	0.7844	0.7101	0.7826	0.6987	0.7890	0.6948	0.7885	0.6424	0.7523	0.6674	0.7721	0.6887	0.7725	**0.7388**	**0.7990**
	Full	50%	0.6947	0.7884	0.6981	0.7897	0.6962	0.7814	0.6912	0.7863	0.6865	0.7746	0.6424	0.7523	0.6630	0.7713	0.6810	0.7672	**0.7307**	**0.7977**
	Full	70%	0.6944	0.7844	0.6920	0.7868	0.6938	0.7806	0.6782	0.7802	0.6652	0.7673	0.6389	0.7538	0.6585	0.7682	0.6757	0.7593	**0.7268**	**0.7941**
	Full	90%	0.6759	0.7784	0.6811	0.7794	0.6846	0.7893	0.6740	0.7751	0.6508	0.7592	0.6179	0.7399	0.6224	0.7530	0.6452	0.7441	**0.7197**	**0.7922**
	Full	Full	0.7237	0.7986	0.7255	0.7966	0.7289	0.7979	0.7246	0.7953	0.7197	0.7941	0.6946	0.7776	0.6988	0.7788	0.7290	0.7983	**0.7436**	**0.8009**
	RGB	AoP	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
CrackPolar	10%	Full	0.7175	0.7912	0.7140	0.7958	0.7049	0.7928	0.7030	0.7854	0.7085	0.7904	0.6718	0.7678	0.6859	0.7757	0.6704	0.7582	**0.7352**	**0.7965**
	30%	Full	0.6958	0.7854	0.6917	0.7814	0.6935	0.7858	0.6971	0.7826	0.6999	0.7825	0.6499	0.7476	0.6593	0.7554	0.5756	0.6877	**0.7314**	**0.7962**
	50%	Full	0.6831	0.7810	0.6869	0.7711	0.6736	0.7768	0.6751	0.7730	0.6728	0.7768	0.6317	0.7445	0.6204	0.7482	0.5331	0.6908	**0.7176**	**0.7858**
	70%	Full	0.6631	0.7550	0.6612	0.7604	0.6605	0.7591	0.6631	0.7559	0.6749	0.7698	0.6375	0.7438	0.6079	0.7281	0.5353	0.6748	**0.7075**	**0.7838**
	90%	Full	0.6499	0.7548	0.6415	0.7519	0.6540	0.7564	0.6593	0.7563	0.6624	0.7636	0.6117	0.7326	0.5762	0.7112	0.4989	0.6660	**0.6969**	**0.7756**
	Full	10%	0.7118	0.7890	0.7162	0.7917	0.7227	0.7916	0.7025	0.7883	0.7077	0.7882	0.6704	0.7685	0.6822	0.7739	0.7154	0.7880	**0.7464**	**0.8054**
	Full	30%	0.7022	0.7914	0.7086	0.7892	0.7094	0.7931	0.6998	0.7843	0.6999	0.7825	0.6574	0.7628	0.6616	0.7696	0.6941	0.7770	**0.7444**	**0.8048**
	Full	50%	0.6938	0.7890	0.7038	0.7845	0.6926	0.7912	0.6901	0.7807	0.6904	0.7826	0.6327	0.7543	0.6644	0.7707	0.6897	0.7752	**0.7448**	**0.8028**
	Full	70%	0.6927	0.7861	0.6981	0.7772	0.6922	0.7857	0.6805	0.7794	0.6788	0.7774	0.6394	0.7552	0.6481	0.7683	0.6774	0.7683	**0.7346**	**0.7994**
	Full	90%	0.6824	0.7795	0.6800	0.7863	0.6854	0.7894	0.6708	0.7685	0.6748	0.7683	0.6105	0.7486	0.6236	0.7537	0.6619	0.7570	**0.7329**	**0.7970**
	Full	Full	0.7240	0.7933	0.7273	0.7906	0.7278	0.7982	0.7219	0.7916	0.7194	0.7975	0.6846	0.7746	0.6929	0.7796	0.7354	0.8010	**0.7488**	**0.8056**

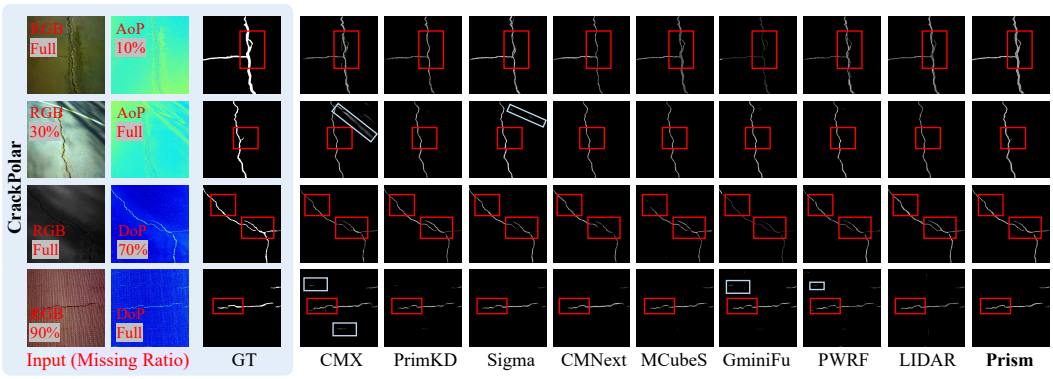


Figure 9: Visual comparison under dual-modal and multi missing settings on the CrackPolar dataset. Red boxes denote key regions; blue denote misidentifications.

Notably, even when modal data is complete, Prism demonstrates a significant advantage, achieving an F1 of 0.7436. This surpasses the strong-performing LIDAR [25] by 2.01% under the same conditions. This underscores the robust adaptive modeling capability of the PPI strategy within the PulseMixer module regarding cross-modal deep topological structures and anisotropic texture information. Simultaneously, the UCPF module, guided by uncertainty estimation, facilitates deep interaction and fusion of feature information across different modalities and hierarchies. Consequently, this generates crack segmentation maps with effectively suppressed background noise and distinct boundaries, fundamentally elevating the segmentation performance.

I.2 Experiments on CrackPolar seven-modal Input

To verify the performance of Prism under extensive multimodal input conditions, we conducted comparative experiments on the CrackPolar [25] dataset. We utilized all seven modalities available in the CrackPolar dataset as input for Prism, including RGB, AoP, DoP, and four typical polarization angle images at 0°, 45°, 90°, and 135°. The AoP modality reflects the spatial distribution of light wave vibration directions, making it sensitive to surface microstructures and orientation variations. The DoP modality describes the purity of light polarization, which is closely related to surface reflection characteristics and roughness. The four polarization angle images record light intensity distributions under four distinct polarization directions, capturing anisotropic reflection features of the surface relative to incident light.

As shown in Table 11, our method consistently achieved the best results across all missing rate settings. For instance, when only the RGB modality was complete, while AoP, DoP, and the four polarization angle modalities were missing with rates of 10%, 30%, and 70%, respectively, Prism

Table 11: Performance comparison under all seven-modal inputs and different missing ratio settings on the CrackPolar dataset. **Purple** and Green denote best and second-best results.

Missing Ratio				CMNeXT		GminiFu		MCubeS		mmsFormer		PWRF		LIDAR		**Ours**	
RGB	AoP	DoP	[0°, 45°, 90°, 135°]	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
Full	10%	30%	70%	0.7317	0.8018	0.6402	0.7512	0.7303	0.7975	0.7213	0.7968	0.7122	0.7849	0.7178	0.7897	**0.7493**	**0.8053**
Full	30%	50%	90%	0.7213	0.7905	0.6671	0.7540	0.7144	0.7928	0.7107	0.7926	0.7117	0.7869	0.7130	0.7863	**0.7470**	**0.8044**
30%	Full	70%	10%	0.7315	0.7984	0.7090	0.7852	0.7252	0.7974	0.7175	0.8005	0.7196	0.7903	0.7200	0.7885	**0.7511**	**0.8091**
10%	70%	Full	30%	0.7380	0.8031	0.7055	0.7869	0.7306	0.8004	0.7131	0.7981	0.7106	0.7881	0.6959	0.7746	**0.7518**	**0.8085**
70%	30%	10%	Full	0.7233	0.7991	0.7031	0.7810	0.7199	0.7999	0.7190	0.7986	0.7192	0.7886	0.7152	0.7880	**0.7540**	**0.8099**
Full	Full	30%	30%	0.7303	0.8058	0.6978	0.7771	0.7218	0.8015	0.7207	0.7970	0.7162	0.7891	0.7173	0.7899	**0.7525**	**0.8093**
Full	30%	Full	30%	0.7195	0.7895	0.7000	0.7757	0.7264	0.7972	0.7233	0.7992	0.7165	0.7903	0.7187	0.7893	**0.7510**	**0.8070**
30%	50%	Full	Full	0.7250	0.8015	0.6923	0.7796	0.7269	0.7997	0.7218	0.7949	0.7190	0.7900	0.7161	0.7882	**0.7504**	**0.8079**
Full	30%	30%	Full	0.7148	0.7995	0.6390	0.7628	0.7132	0.7976	0.7222	0.7988	0.7184	0.7894	0.7153	0.7871	**0.7544**	**0.8103**
30%	Full	Full	30%	0.7242	0.7948	0.7179	0.7919	0.7253	0.7967	0.7206	0.7947	0.7171	0.7899	0.7152	0.7839	**0.7524**	**0.8088**
Full	Full	Full	90%	0.7158	0.7940	0.6908	0.7775	0.7018	0.7889	0.7127	0.7884	0.7079	0.7813	0.7150	0.7869	**0.7524**	**0.8088**
90%	Full	Full	Full	0.7340	0.8016	0.7248	0.7966	0.7210	0.7984	0.7101	0.7947	0.7156	0.7884	0.7184	0.7879	**0.7529**	**0.8079**
Full	90%	Full	Full	0.7196	0.7998	0.7021	0.7776	0.7132	0.7965	0.7183	0.7960	0.7254	0.7927	0.7231	0.7904	**0.7518**	**0.8078**
Full	Full	90%	Full	0.7214	0.7993	0.6096	0.7424	0.7182	0.7996	0.7190	0.7978	0.7178	0.7900	0.7059	0.7799	**0.7531**	**0.8094**
10%	30%	70%	90%	0.7118	0.7972	0.6858	0.7711	0.7060	0.7867	0.7048	0.7888	0.6672	0.7694	0.6804	0.7546	**0.7431**	**0.8021**
30%	50%	90%	10%	0.7235	0.7923	0.6916	0.7798	0.7158	0.7862	0.6954	0.7890	0.6746	0.7762	0.6959	0.7781	**0.7441**	**0.8083**
90%	70%	30%	10%	0.7256	0.7958	0.6899	0.7771	0.7124	0.7997	0.7167	0.7986	0.7205	0.7885	0.7168	0.7888	**0.7490**	**0.8066**
Full	Full	Full	Full	0.7391	0.8029	0.7051	0.7848	0.7383	0.8008	0.7264	0.7998	0.7219	0.7921	0.7451	0.8037	**0.7569**	**0.8103**

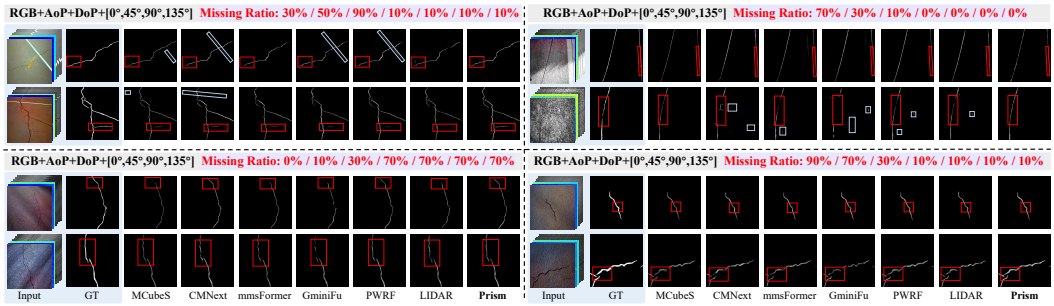


Figure 10: Visual comparison under seven-modal and multimissing settings on the CrackPolar dataset. Red boxes denote key regions; blue denote misidentifications.

Table 12: Complexity comparison of seven-modal 512 × 512 input.

Method	Year	FLOPs↓	Params↓	Size↓
MCubeS [19]	CVPR 2022	550.48G	410.37M	4496MB
CMNeXT [54]	CVPR 2023	**50.92G**	57.69M	660MB
mmsFormer [36]	OJSP 2024	93.50G	86.68M	1034MB
GminiFu [14]	ICML 2024	51.75G	29.27M	358MB
PWRF [29]	IJCV 2025	451.58G	710.67M	7024MB
LIDAR [25]	MM 2025	116.68G	18.68M	270MB
Prism (Ours)	N/A	101.13G	**8.83M**	**128MB**

achieved an F1 of 0.7493 and an mIoU of 0.8053. When the four polarization angle modalities were complete, but the RGB, AoP, and DoP modalities had missing rates of 70%, 30%, and 10%, respectively, Prism attained an F1 of 0.7540, outperforming the second-best method, MCubeS [19], by 4.73%. Furthermore, in extreme scenarios where all seven modalities exhibited varying degrees of missing data, specifically, with missing rates of 90%, 70%, 30%, and 10% for RGB, AoP, DoP, and the four polarization angle modalities, Prism still achieved an F1 of 0.7490 and an mIoU of 0.8066. These results represent improvements of 3.23% and 1.35%, respectively, over the second-best method, CMNeXT [54]. Additionally, as illustrated in Figure 10, Prism maintained continuous segmentation results in critical regions. These findings indicate that Prism can leverage the MoRe module to effectively restore and complete missing modal data and features even under extreme modality miss. Moreover, it can utilize PulseMixer and UCPF to adaptively and efficiently perceive and fuse topological and textural features from different modalities and hierarchical levels, thereby generating high-quality crack segmentation maps.

As listed in Table 12, we compared the complexity of Prism with other methods under the seven-modal input setting. Although Prism did not achieve the lowest FLOPs in this configuration, it required only 8.83M parameters and a model size of 128MB. This corresponds to reductions of 52.73% and 52.59%, respectively, compared to LIDAR [25], a multimodal crack segmentation method known for its lightweight design. This demonstrates that our method possesses excellent edge deployment capabilities even when facing massive data inputs. This efficiency is attributed to the sparse gating

Table 13: Performance comparison under three-modal and extreme missing settings. **Purple** and Green denote best and second-best results.

Missing Ratio			GminiFusion		MCubeS		mmsFormer		PWRF		CMNeXT		LIDAR		**Prism**	
RGB	AoP	DoP	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU
Full	Full	100%	0.6406	0.7453	0.7018	0.7885	0.6978	0.7892	0.6752	0.7740	0.7112	0.7928	0.7065	0.7840	**0.7311**	**0.7946**
Full	100%	Full	0.6368	0.7470	0.7030	0.7917	0.6970	0.7835	0.6827	0.7761	0.7139	0.7944	0.7097	0.7858	**0.7324**	**0.8013**
100%	Full	Full	0.6131	0.7424	0.6881	0.7816	0.6839	0.7822	0.6766	0.7753	0.6926	0.7854	0.6886	0.7717	**0.7300**	**0.7947**
Full	Full	Full	0.6920	0.7794	0.7163	0.7936	0.7265	0.7952	0.7063	0.7837	0.7124	0.7875	0.7353	0.7976	**0.7510**	**0.8066**

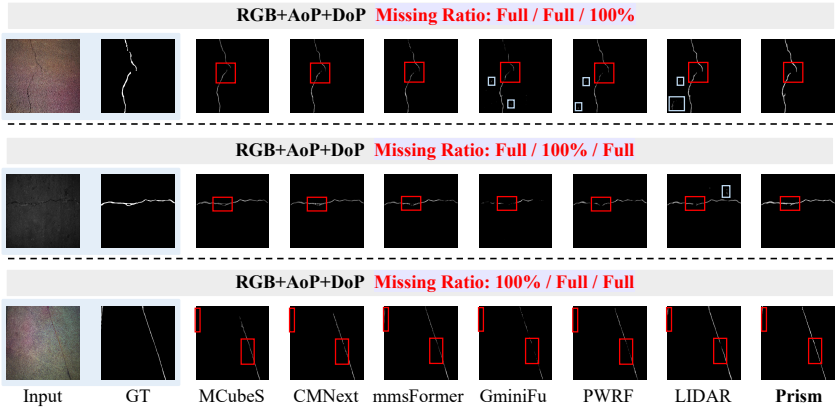


Figure 11: Visual comparison under seven-modal and extreme missing settings on the CrackPolar dataset. Red boxes denote key regions; blue denote misidentifications.

mechanism within PulseMixer, which significantly reduces the computational overhead required by the model while maintaining robust feature representation capabilities.

I.3 Experiments with Extreme Missing Modalities

We evaluated the robustness of Prism under extreme conditions where one modality is entirely unavailable on CrackPolar dataset. As shown in Table 13 and Figure 11, Prism consistently outperforms other methods across all single-modality dropout settings.

With RGB and AoP complete but DoP entirely missing, Prism achieves an F1 of 0.7311 and an mIoU of 0.7946, surpassing the second-best CMNeXT [54] by 2.80% and 0.23%, indicating that implicit structural alignment successfully registers incomplete polarization cues with complete RGB features for high-precision fusion. Similarly, when AoP is 100% missing, Prism maintains superior performance with an F1 of 0.7324, exceeding CMNeXT [54] by 2.59%, suggesting that the sparse gating mechanism effectively filters noise from the remaining DoP channel to compensate for the lost polarization dimension.

In the most challenging setting where the primary RGB modality is 100% missing, leaving only AoP and DoP, Prism still attains an F1 of 0.7300 and an mIoU of 0.7947. This performance exceeds CMNeXT [54] by 5.40% in F1 and 1.18% in mIoU, producing clear segmentation boundaries solely from polarization data. This result suggests that MoRe and PaKD align the manifolds of the remaining polarization modalities toward the complete feature space, significantly reducing degradation caused by the absence of color texture information. Even when all modalities are fully available, Prism achieves an F1 of 0.7510 and an mIoU of 0.8066, surpassing LIDAR [25] by 2.14% and 1.13%, validating that PulseMixer and UCPF preserve physical attributes and geometric structure to maximize multimodal synergy.

I.4 Experiments on Single-modal Input

To validate the performance of Prism under the single RGB modality, we conducted comparative experiments against other SOTA single-modal crack segmentation methods on the CrackPolar, IRTCrack, and CrackDepth datasets. As presented in Table 14, Prism consistently achieved superior segmentation performance while maintaining the lowest computational overhead when provided with only RGB input. On the CrackPolar and CrackDepth datasets, which are characterized by diverse

Table 14: Performance comparison under single RGB modal input.

Method	CrackPolar		IRTCrack		CrackDepth		Params↓	FLOPs↓	Size↓
	F1	mIoU	F1	mIoU	F1	mIoU			
CrackFormer [22]	0.7104	0.7729	0.8204	0.8185	0.8059	0.8393	4.55M	81.85G	55MB
SimCrack [13]	0.7125	0.7889	0.8142	0.8242	0.7940	0.8291	29.58M	286.62G	225MB
MambaIR [9]	0.7063	0.7777	0.8297	0.8294	0.7912	0.8254	10.34M	47.32G	79MB
CSMamba [26]	0.6428	0.7342	0.7909	0.7925	0.7238	0.7753	35.95M	145.85G	233MB
PlainMamba [48]	0.7161	0.7824	0.8294	0.8277	0.8013	0.8306	16.72M	73.86G	96MB
SCSegamba [24]	0.7173	0.7925	0.8214	0.8284	0.8008	0.8342	2.80M	18.16G	37MB
Prism (Ours)	**0.7426**	**0.8019**	**0.8469**	**0.8509**	**0.8250**	**0.8470**	**1.37M**	**16.58G**	**19MB**

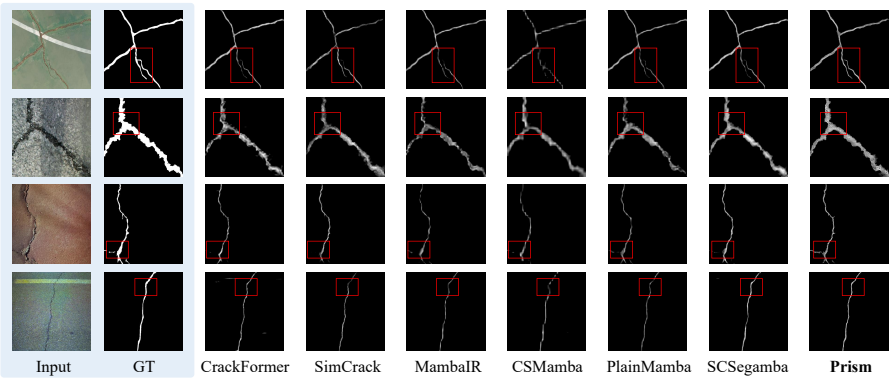


Figure 12: Visual comparison under signle RGB modal. Red boxes denote key regions.

scenes, complex lighting, and severe background interference, Prism achieved mIoU of 0.8019 and 0.8470, respectively, surpassing SCSegamba [24]. Notably, on CrackDepth, the improvement in the F1 exceeded 3.02%. Furthermore, as illustrated in Figure 12, Prism produced more continuous and clearer segmentation results in fine-grained and heavily interfered crack regions compared to other methods. This demonstrates that the PPI strategy within PulseMixer can adaptively perceive deep crack topological and morphological information, thereby strengthening the correlation between crack pixels to yield more detailed segmentation results without reliance on additional modalities. On the IRTCrack dataset, where crack regions are larger with blurred boundaries, Prism still achieved an F1 of 0.8469 and an mIoU of 0.8509, outperforming SCSegamba by 3.10% and 2.72%, respectively. This indicates that even with single-modal features, PulseMixer and UCPF can effectively perceive and fuse deep spatial and morphological attributes within cross-level crack image features, generating high-quality segmentation maps that effectively suppress background noise.

I.5 Experiments on Other Domain Datasets

To further verify the generalization capability of Prism beyond the primary crack segmentation benchmarks, we conducted additional experiments on three datasets from different application domains, including VT1000 [39] for RGB-T salient object detection, MEMO [41], repurposed for retinal vessel segmentation using its EMA vessel masks, and KUST4K [32] for UAV-based road segmentation. These datasets differ substantially from structural crack inspection in terms of scene distribution, target morphology, and sensor configuration, thereby providing a broader evaluation of the proposed implicit structural alignment and pulse propagation mechanisms. For VT1000, we used RGB and thermal modalities with RGB missing at 10% and infrared missing at 50%. For MEMO, we used EMA and OCTA modalities with missing rates of 10% and 30%, respectively. For KUST4K, RGB was kept complete while the TIR modality was missing at 70%. All methods were evaluated under the same missing-modality settings for fair comparison.

As shown in Table 15 and Figure 13, Prism consistently achieves the best performance across all three external datasets. On VT1000 [39], Prism obtains an F1 of 0.8442 and an mIoU of 0.8577, surpassing the second-best MCubeS by 1.87% and 3.10%, respectively. This indicates that the proposed framework can effectively handle RGB-T salient object segmentation even when both visible and thermal modalities are partially missing.

Table 15: Performance comparison on other domain datasets. **Purple** and Green denote best and second-best results.

Method	VT1000 [39]		MEMO [41]		KUST4K [32]	
	F1	mIoU	F1	mIoU	F1	mIoU
CMX	0.7838	0.7882	0.5092	0.6393	0.8875	0.7708
CMNeXT	0.8225	0.8261	0.5089	0.6405	0.8928	0.7945
MCubeS	0.8287	0.8319	0.5123	0.6417	0.8902	0.7892
Sigma	0.6269	0.6776	0.4968	0.6333	0.8884	0.7717
VRWKV	0.8165	0.8272	0.5065	0.6424	0.8970	0.7896
Prism (Ours)	**0.8442**	**0.8577**	**0.5359**	**0.6512**	**0.9075**	**0.8006**

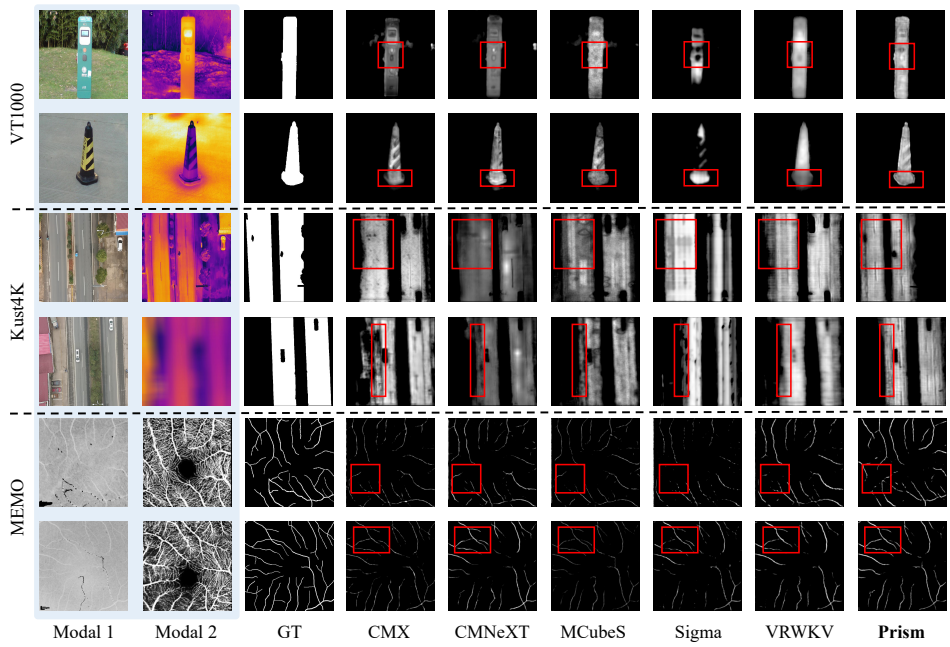


Figure 13: Visual comparison analysis on other domain datasets. Red boxes denote key regions.

On MEMO [41], Prism achieves an F1 of 0.5359 and an mIoU of 0.6512. Compared with the strongest competing results, Prism improves the F1 over MCubeS by 4.61% and the mIoU over VRWKV by 1.37%. These results suggest that the implicit structural alignment learned by MoRe and PaKD can also benefit medical vessel segmentation, where the target regions are thin, sparse, and highly sensitive to local discontinuities.

On KUST4K [32], Prism reaches an F1 of 0.9075 and an mIoU of 0.8006 under UAV-based road segmentation with severe TIR missing. It outperforms the best competing F1 result from VRWKV by 1.17% and the best competing mIoU result from CMNeXT by 0.77%. This further demonstrates that PulseMixer and UCPF can preserve elongated structural regions and suppress background interference in remote sensing scenarios.

These cross-domain results show that Prism is not restricted to the three main crack datasets. Although the primary focus of this work is multimodal crack segmentation, the proposed implicit structural alignment, topology-aware pulse propagation, and uncertainty-guided fusion can transfer to other multimodal segmentation tasks involving salient, vessel-like, or elongated foreground structures.

J More Ablation and Sensitivity Studies

J.1 Ablation Studies on Different Components

Table 16 presents a systematic decomposition of the Prism architecture to isolate the contributions of MoRe, GPM, and Sparse Gating (SG). Analyzing the performance degradation upon removing MoRe reveals a significant drop of 1.31% in F1 and 0.98% in mIoU. This empirical evidence validates Theorem A.1, confirming that the student model heavily relies on the implicit structural alignment

with the SAM2 teacher to navigate the feature manifold. Without MoRe, the model loses the topological priors distilled from the foundation model and struggles to reconstruct missing features in occlusion scenarios. Crucially, adding MoRe incurs a negligible cost of only 0.01M parameters and 0.12G FLOPs, confirming the superiority of our Prompt-as-Knowledge Distillation strategy which successfully transfers the heavy reasoning capability of the teacher into the lightweight student without inheriting the computational burden.

GPM serves as the structural backbone of the network, and its removal causes a decline of 0.91% in F1 and 0.78% in mIoU. This degradation is directly attributed to the reintroduction of Manifold Distortion as defined in Theorem B.1. Without GPM, the sequence modeling reverts to standard raster scanning or isotropic diffusion, which severs the intrinsic connectivity of crack structures and introduces false proximity errors. The associated cost is a justifiable investment for establishing a geometry-aware scanning order that preserves the geodesic continuity of thin, curvilinear defects.

The most critical drop in F1 of 1.37% occurs when SG is removed. This strongly supports Theorem B.2, indicating that the base linear attention suffers from the Static Parameterization Bottleneck. SG alleviates this by introducing data-dependent non-linearity, effectively increasing the rank of the operator space via polyhedral decomposition. Furthermore, SG acts as a dynamic filter that suppresses background noise variance, and without it, the model loses its discriminative capacity to separate faint crack signals from noisy industrial backgrounds. The minimal configuration yields the lowest segmentation performance, demonstrating that distillation alone is insufficient without a capable architecture to absorb the knowledge. The full Prism model achieves the optimal trade-off, where GPM ensures structural integrity, SG expands representational capacity, and MoRe provides semantic guidance, all while maintaining a lightweight profile suitable for edge deployment.

Table 16: Performance comparison of different components.

MoRe	GPM	SG	F1	mIoU	Params↓	FLOPs↓	Size↓
✓	✗	✗	0.7315	0.7960	**3.66M**	**41.76G**	**52MB**
✗	✓	✗	0.7288	0.7979	3.83M	44.54G	55MB
✗	✗	✓	0.7287	0.7949	3.68M	41.84G	53MB
✓	✓	✗	0.7341	0.7991	3.83M	44.64G	56MB
✓	✗	✓	0.7376	0.7987	3.68M	41.93G	53MB
✗	✓	✓	0.7347	0.7971	3.85M	44.71G	56MB
✓	✓	✓	**0.7443**	**0.8049**	3.86M	44.77G	56MB

J.2 Ablation Studies within UCPF

To comprehensively evaluate the impact of different components within UCPF on performance, we conducted ablation experiments on its constituent parts. Specifically, Enh denotes the Enhance stage designed for feature enhancement during the input phase. UA represents the Uncertainty Aware stage utilized for cross-modal uncertainty perception. GP signifies the Gate Propagation stage employed for gated propagation and feature selection.

Table 17: Ablation study on different components in UCPF.

Enhance	CMA	FinalPro	F1	mIoU	Params↓	FLOPs↓	Size↓
✓	✗	✗	0.7311	0.8005	**3.75M**	43.43G	**54MB**
✗	✓	✗	0.7268	0.7940	3.85M	43.96G	56MB
✗	✗	✓	0.7323	0.8026	3.76M	**42.70G**	55MB
✓	✓	✗	0.7405	0.8032	3.86M	44.76G	56MB
✓	✗	✓	0.7390	0.8034	3.76M	43.50G	55MB
✗	✓	✓	0.7389	0.8044	3.85M	44.02G	56MB
✓	✓	✓	**0.7443**	**0.8049**	3.86M	44.77G	56MB

As presented in Table 17, the full implementation of UCPF enables Prism to achieve optimal performance, yielding F1 and mIoU of 0.7443 and 0.8049, respectively. When the Enh stage is omitted, the model performance experiences a slight decline, resulting in an F1 of 0.7389 and an mIoU of 0.8044. This indicates that performing channel scaling and extracting deep spatial information immediately upon the input of multi-scale crack features from diverse modalities and hierarchical levels into UCPF is crucial for the model to perceive salient information within crack regions.

When the UA stage is excluded from UCPF, the F1 decreases to 0.7390, representing a 0.71% reduction compared to the complete UCPF. This demonstrates that the UA stage leverages uncertainty estimation to effectively weight and denoise features from different modalities while maintaining low computational overhead, as the parameter count differs by only 0.1M. Consequently, it shields the

model from interference introduced by modality miss or alignment errors, thereby achieving superior segmentation results.

Notably, when the GP stage is removed from UCPF, the model still achieves an F1 of 0.7405, yet this represents a performance loss of 0.51% compared to the full model. This suggests that the GP stage utilizes gating mechanisms and topological propagation to further refine boundary responses of features and enhance the connectivity of crack structures. Collectively, the three components of UCPF complement one another, jointly contributing to the superior performance of Prism in multimodal crack segmentation tasks.

J.3 Ablation Study on Different Backbones

To validate the necessity and effectiveness of PulseMixer, we replaced it with several representative lightweight backbones while keeping the remaining components unchanged.

Table 18: Ablation study on different backbone.

Variants	F1	mIoU	Params↓	FLOPs↓	Size↓
MobileNetV4 [34]	0.7065	0.7806	**3.43M**	151.37G	**40MB**
EfficientViT [28]	0.7124	0.7876	4.20M	182.26G	49MB
VMamba [30]	0.7169	0.7827	6.07M	112.88G	72MB
VRWKV [8]	0.7087	0.7823	14.74M	188.86G	187MB
PulseMixer(Ours)	**0.7443**	**0.8049**	3.86M	**44.77G**	56MB

As shown in Table 18, replacing PulseMixer with generic lightweight backbones leads to consistent performance degradation. Compared with VMamba [30], which achieves the second-best F1 score, our PulseMixer-based design improves F1 from 0.7169 to 0.7443, yielding a relative gain of 3.82%. Compared with EfficientViT [28], which achieves the second-best mIoU, Prism improves mIoU from 0.7876 to 0.8049, corresponding to a relative gain of 2.20%. This indicates that the performance improvement does not merely come from using a compact backbone, but from the topology-aware pulse propagation and sparse gated recurrent modeling within PulseMixer.

Compared with VRWKV [8], Prism improves F1 by 5.02% and mIoU by 2.89%, while reducing Params, FLOPs, and model size by 73.81%, 76.29%, and 70.05%, respectively. This demonstrates that directly adopting a standard vision RWKV backbone is insufficient for multimodal crack segmentation, since its conventional token mixing lacks explicit structural ordering and adaptive propagation along crack topology. Compared with VMamba, Prism further reduces Params by 36.41%, FLOPs by 60.34%, and model size by 22.22%, while achieving higher segmentation accuracy. These results verify that PulseMixer achieves a better accuracy-efficiency trade-off by combining PPI-based skeleton-aware serialization with sparse gated feature modulation.

Although MobileNetV4 [34] has the lowest parameter count and model size, its F1 and mIoU are only 0.7065 and 0.7806, respectively, indicating that the tested lightweight backbone struggles to capture long-range structural continuity under missing modalities. EfficientViT [28] also suffers from high computational cost, requiring 182.26G FLOPs while still trailing Prism by 4.48% in F1 and 2.20% in mIoU. Overall, the backbone replacement study confirms that PulseMixer is not a replaceable generic feature extractor, but an essential component for preserving anisotropic crack topology and maintaining robust segmentation under incomplete multimodal inputs.

J.4 Ablation Study on Different Fusion Methods

To validate the effectiveness of UCPF, we replaced it with direct addition, channel-wise concatenation, an adapted DFormerV2 depth-as-geometry-prior baseline [51], FreqFusion [3], and StitchFusion [17].

Table 19: Ablation study on fusion variants.

Variants	F1	mIoU	Params↓	FLOPs↓	Size↓
Add	0.7230	0.7916	**3.75M**	**42.59G**	**55MB**
Concat	0.7299	0.7895	4.01M	46.03G	58MB
DFormerV2 [51]	0.7235	0.7898	8.56M	48.56G	109MB
FreqFusion [3]	0.7297	0.7942	3.93M	45.52G	59MB
StitchFusion [17]	0.7300	0.7948	4.39M	51.02G	62MB
UCPF(Ours)	**0.7443**	**0.8049**	3.86M	44.77G	56MB

As shown in Table 19, replacing UCPF with generic fusion strategies consistently leads to performance degradation. Compared with StitchFusion, which achieves the second-best F1 and mIoU, UCPF improves F1 from 0.7300 to 0.7443 and mIoU from 0.7948 to 0.8049, corresponding to relative gains of 1.96% and 1.27%, respectively. This indicates that the advantage of UCPF does not simply come from combining

multi-level features, but from uncertainty-guided reliability estimation and connectivity-aware propagation.

Compared with direct addition, UCPF improves F1 by 2.95% and mIoU by 1.68%, while introducing only 2.93% more parameters, 5.12% more FLOPs, and 1.82% larger model size. This demonstrates that the additional uncertainty-aware fusion and gated propagation bring clear segmentation gains with only marginal complexity overhead. Compared with channel-wise concatenation, UCPF improves F1 by 1.97% and mIoU by 1.95%, while reducing Params, FLOPs, and model size by 3.74%, 2.74%, and 3.45%, respectively. This further shows that simple feature stacking cannot effectively distinguish reliable modal cues from noisy or missing regions. Although the adapted DFormerV2 baseline [51], FreqFusion [3], and StitchFusion [17] introduce stronger designs than Add or Concat, they still underperform UCPF. In particular, the adapted DFormerV2 baseline requires 8.56M parameters and 109MB model size, yet its F1 is only 0.7235, indicating that this depth-guided design is not necessarily suitable for lightweight multimodal crack segmentation. Compared with StitchFusion, UCPF reduces Params, FLOPs, and model size by 12.07%, 12.25%, and 9.68%, respectively, while achieving higher segmentation accuracy. Overall, these results verify that UCPF is an essential fusion module for suppressing unreliable modality responses, repairing topological discontinuities, and maintaining a favorable accuracy-efficiency trade-off under incomplete multimodal inputs.

J.5 Ablation Study on Different Missing-Modality Degradation Forms

To further examine the robustness of Prism beyond zero-fill, we evaluate both alternative missing-signal representations and limited cross-modal misregistration on CrackPolar with RGB fully available and both AoP and DoP subject to a 50% missing rate. Gaussian noise is sampled in the 8-bit range from $\mathcal{N}(128, 50^2)$, block occlusion zeros each 64×64 patch with probability 0.6, and signal attenuation scales the affected modality by a factor sampled from $[0.05, 0.20]$; each variant replaces zero-fill during training and testing. For misregistration, translations of ± 10 pixels or rotations of $\pm 5^\circ$ are applied to the available AoP and DoP inputs only at inference without adaptation, and the two directions are averaged.

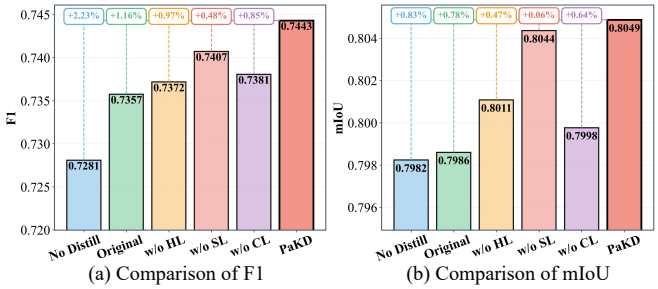
Table 20: Robustness to missing-signal forms and cross-modal misregistration. Blue bold and green underlined entries denote best and second-best results.

Setting	F1	mIoU
Zero-fill	0.7443	0.8049
Gaussian noise	0.7445	0.8026
Block occlusion	**0.7452**	**0.8066**
Signal attenuation	0.7429	0.8031
Translation (± 10 px)	0.7427	<u>0.8052</u>
Rotation ($\pm 5^\circ$)	<u>0.7450</u>	0.8032

As shown in Table 20, F1 remains within 0.7427–0.7452 and mIoU within 0.8026–0.8066, corresponding to ranges of only 0.25 and 0.40 percentage points. The comparable results under noise, occlusion, attenuation, translation, and rotation indicate that Prism does not depend on the numerical signature of zero-fill and retains stable performance under limited sensor misregistration.

J.6 Controlled Distillation and Prompt Studies

We first isolate the effect of PaKD from teacher capacity and student architecture. All variants use the same CrackPolar split, with RGB fully available and both AoP and DoP subject to a 50% missing rate; where applicable, the teacher input and training budget are held fixed. Standard Distill matches teacher predictions without topology prompts or the structural objectives used by PaKD.



(a) Comparison of F1 (b) Comparison of mIoU

Figure 14: Comparison of different distillation settings. HL represents $\mathcal{L}_{hm}$, SL represents $\mathcal{L}_{struc}$, CL represents $\mathcal{L}_{cont}$.

Table 21: Controlled comparison of student architectures and distillation supervision. Blue bold and green underlined entries denote best and second-best results.

Student	Supervision	F1	mIoU
LIDAR	No Distill	0.7153	0.7832
LIDAR	Standard Distill, SAM2-Base+	0.7201	0.7955
Prism	No Distill	0.7281	0.7982
Prism	Standard Distill, SAM2-Base+	0.7307	0.7948
Prism	GT Heatmap	0.7375	0.7996
Prism	Full PaKD, SAM1-ViT-B	0.7419	0.8023
Prism	Full PaKD, SAM2-Tiny	<u>0.7431</u>	**0.8054**
Prism	Full PaKD, SAM2-Base+	**0.7443**	<u>0.8049</u>

As shown in Table 21, standard distillation improves LIDAR from 0.7153/0.7832 to 0.7201/0.7955 in F1/mIoU, confirming that the teacher itself is beneficial under the controlled protocol. With the same standard supervision, Prism attains a higher F1 of 0.7307. Full PaKD further reaches 0.7443/0.8049, exceeding both distilled LIDAR and standard-distilled Prism. Across the three SAM teachers, F1 differs by at most 0.24 percentage points, indicating that the gain is not tied to a single teacher scale.

We next perturb the training prompts while keeping the same split and missingness setting. Fully Random samples both coordinates and labels without using the ground truth, image, or student prediction, whereas GT+noise perturbs the ground-truth-derived coordinates with 2D Gaussian noise whose standard deviation is σ pixels.

Table 22: Robustness to training-prompt construction. Blue bold and green underlined entries denote best and second-best results.

Prompt Setting	F1	mIoU
No Distill	0.7281	0.7982
Fully Random	0.7336	0.7998
GT+noise ($\sigma = 5$)	<u>0.7438</u>	<u>0.8042</u>
GT+noise ($\sigma = 10$)	0.7421	0.8031
GT+noise ($\sigma = 20$)	0.7389	0.8008
Full PaKD	**0.7443**	**0.8049**

Table 22 shows that even fully random prompts improve over No Distill, while moderate perturbations preserve most of the full PaKD gain. Performance decreases gradually as prompt noise increases, which supports the use of topology-aware prompts while showing that PaKD is not brittle to prompt-coordinate perturbations.

J.7 Sensitivity Studies on Different Patch Size

To verify the impact of the Patch Size within Prism on model performance and computational efficiency, we conducted sensitivity experiments on the CrackPolar dataset.

As illustrated in Table 23 and Figure 15, the selection of Patch Size exhibits a significant nonlinear influence on model performance. When the Patch Size is set to 4, Prism achieves an optimal balance between performance and efficiency, attaining peak F1 and mIoU of 0.7443 and 0.8049, respectively, while maintaining a reasonable computational overhead of 44.77G. When the Patch Size is reduced to 2, although the model can capture finer local textures, the F1 remains nearly comparable to the optimal setting. However, the computational cost surges to 83.34G, representing an increase of approximately 85.90% compared to a Patch Size of 4, and the model size increases slightly. This indicates that excessive refinement of patches does not yield substantial performance gains but instead results in a waste of computational resources.

Table 23: Sensitivity study on different patch sizes.

Patch Size	F1	mIoU	Params↓	FLOPs↓	Size↓
2	0.7442	0.7991	**3.79M**	83.34G	64MB
4	**0.7443**	**0.8049**	3.86M	44.77G	**56MB**
8	0.7185	0.7899	5.41M	25.98G	69MB
16	0.6847	0.7638	5.63M	13.72G	69MB
32	0.6509	0.7436	5.60M	**8.07G**	70MB

Conversely, as the Patch Size increases, the computational overhead of the model decreases significantly, yet the segmentation performance suffers a sharp decline. Specifically, when the Patch Size is set to 8, the F1 drops to 0.7185, a decrease of 3.47% compared to the optimal setting. When the Patch Size is further increased to 32, the F1 falls to merely 0.6509, corresponding to a substantial reduction of 12.54%. This demonstrates that for slender and discontinuous targets such as cracks, an excessively large Patch Size leads to severe loss of detailed information, making it difficult for the model to effectively perceive the topological structure of minute cracks. Therefore, we selected a Patch Size of 4 for the final scheme to ensure high-precision segmentation while maintaining low computational overhead.

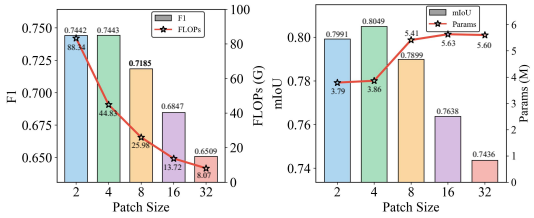


Figure 15: Comparison of performance when the patch size increases.

J.8 Sensitivity Study on PulseMixer Layers

We utilized only four PulseMixer blocks within Prism. To validate the necessity of this configuration, we conducted sensitivity experiments regarding varying numbers of PulseMixer layers.

As illustrated in Table 24 and Figure 16, Prism achieves peak performance in terms of F1 and mIoU scores when the number of PulseMixer layers is set to 4, attaining values of 0.7443 and 0.8049 respectively. It is worth noting that as the number of PulseMixer layers increases, the parameter count, computational volume, and model size of Prism exhibit a significant upward trend. Conversely, the F1 and mIoU display a trend of initial improvement followed by a decline.

Table 24: Sensitivity study on layer numbers.

Layer Num	F1	mIoU	Params↓	FLOPs↓	Size↓
2	0.7381	0.7980	**1.27M**	**17.69G**	**27MB**
4	**0.7443**	**0.8049**	3.86M	44.77G	56MB
8	0.7431	0.8028	7.77M	82.18G	112MB
16	0.7350	0.7947	15.59M	159.31G	224MB
32	0.7179	0.7881	31.23M	313.58G	449MB

When the number of PulseMixer layers becomes excessively high, reaching 32, the F1 and mIoU scores drop to 0.7179 and 0.7881. These figures represent reductions of 3.55% and 2.09% compared to the configuration utilizing only four PulseMixer layers. Simultaneously, the parameter count, computational cost, and model size escalate to 31.23M, 313.58G, and 449MB respectively. This indicates that employing an excessive number of network layers generates significant redundancy regarding morphological and textural information across different modalities. This induces overfitting and causes critical features to be blurred and smoothed during deep propagation. Such phenomena weaken the capability of the model to model spatial hierarchies and morphological characteristics of crack details. Consequently, selecting four PulseMixer layers represents the optimal balance between ensuring high performance feature extraction and maintaining low computational overhead.

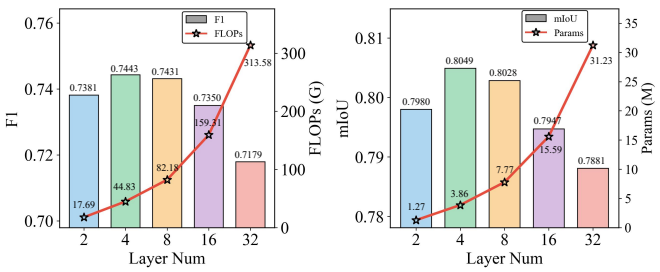


Figure 16: Comparison of performance when the number of layers increases.

K Real-World Deployment

To verify the deployment capability and efficiency of Prism in real-world industrial production scenarios, we constructed a real-world deployment system based on UAV-ground station collaboration and conducted field comparative tests against existing lightweight segmentation methods.

Specifically, our hardware deployment architecture primarily comprises two core modules: the aerial acquisition module and the edge computing module. As illustrated in Figure 17, the aerial acquisition module employs a DJI Mini UAV equipped with a high-definition visual sensor, responsible for video stream data acquisition in complex scenarios. The edge computing module is mounted on an unmanned ground vehicle, with the core computing platform being the NVIDIA Jetson Orin NX Super edge intelligent computing card running the Ubuntu 22.04 operating system.

In the real-world deployment experiment, both the UAV and the intelligent vehicle were placed in an outdoor environment, establishing a low-latency real-time communication link via a high-bandwidth local area network. During the experiment, the UAV maintained a constant cruising speed of 0.2 m/s, performing continuous video acquisition of crack regions on walls or the ground and transmitting the raw video stream back to the edge computing module in real time. Subsequently, the trained Prism model deployed on the Jetson platform performed real-time frame-by-frame inference on the received unknown video stream data, instantly generating high-precision crack segmentation results.

To further evaluate the practical performance of the model, we also deployed pre-trained weights of other mainstream lightweight methods on the edge intelligent computing platform for real-time inference comparison. In the simulation test, the RGB was set to remain complete, while the missing rate of depth modality data reached as high as 90%.

As shown in Figure 18, compared with other segmentation methods, Prism demonstrated significant advantages when processing unfamiliar multimodal video stream data. The generated segmentation masks exhibited clear edges, accurately capturing fine crack branches and effectively suppressing background noise in complex environments. The training-time PaKD mechanism equips MoRe to complete high-missing-rate modalities. In this setting, PaKD uses the available RGB features as structured knowledge prompts during training to guide missing-modality reconstruction through knowledge distillation. This mechanism aligns the clear crack topology and texture information from the available RGB image with the latent feature space, thereby enabling the completed features to retain semantic consistency and geometric integrity under the extreme condition of 90% depth data miss. All teacher and prompt branches are removed at inference.

PulseMixer achieved precise adaptive insight and perception of critical crack regions within different modal data. By simulating pulse propagation characteristics, it effectively enhanced the connectivity response of cracks. Finally, the UCPF module performed efficient and deep interaction and fusion of crack feature maps from different modalities, utilizing uncertainty guidance to shield against noise interference during the modality complementation process.

In contrast, Gminifu [14] failed to effectively suppress irrelevant background texture noise, causing confusion between crack regions and background interference, which resulted in a large number of false positives. mmsFormer [36] exhibited serious false negative problems, missing a substantial amount of critical crack texture details, which indicates a lack of precise crack perception capability on unseen stranger data. Even the LIDAR [25], specifically designed for multimodal crack segmentation, performed unsatisfactorily when facing stranger video streams with high missing rates. It failed to

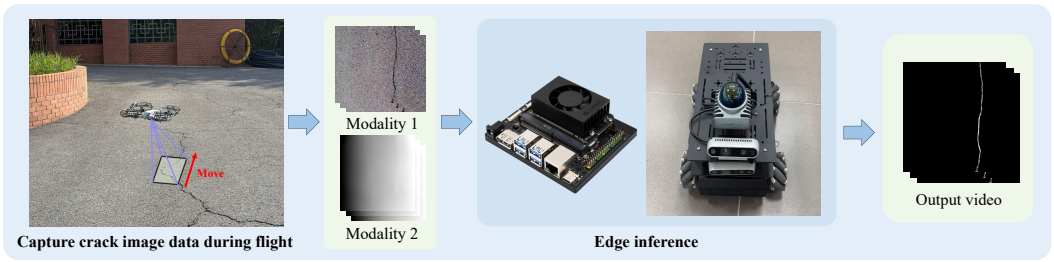


Figure 17: The process of deployment in the real world. The aerial acquisition module is equipped with a high-definition visual sensor, responsible for video stream data acquisition; the edge computing module is mounted on the intelligent vehicle,

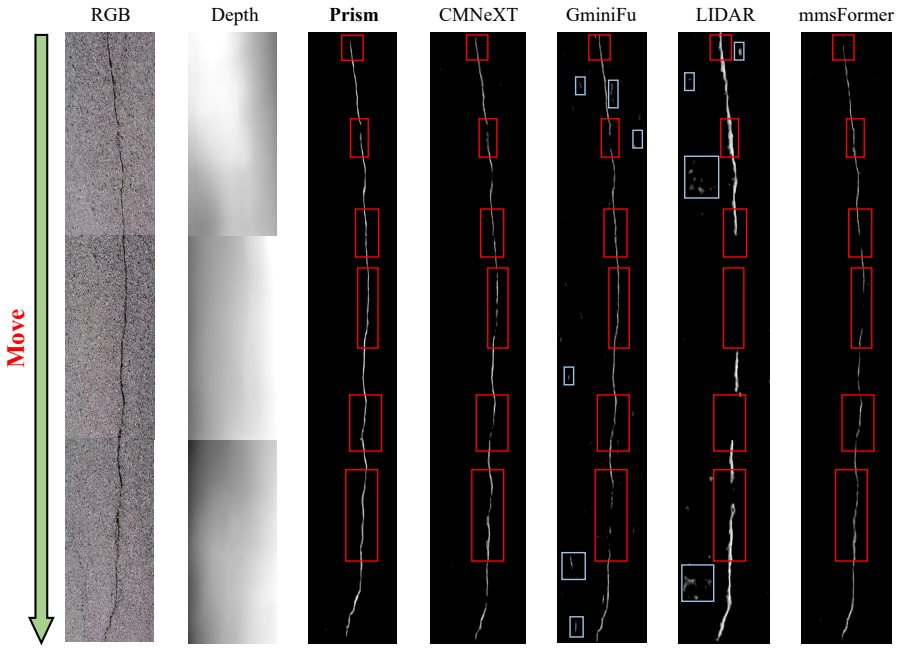


Figure 18: Performance comparison on real videos. Red boxes denote key regions; blue denote misidentifications.

achieve accurate segmentation particularly in critical fine crack connection areas, producing obvious discontinuities and omissions.

Therefore, Prism can still achieve robust and excellent segmentation results on stranger video stream data accompanied by severe modality miss, fully verifying its ability to adapt well and be applied in real-world industrial production processes.

L Reproducibility Details

For all experiments, each dataset was split into training, validation, and test sets with a ratio of 7:1:2. The batch size was set to 1 for all training runs, and the random seed was fixed to 42 for reproducibility. All compared methods were trained and evaluated under the same data split protocol and random seed setting for fair comparison.

M Broader Impacts

This work aims to improve multimodal crack segmentation under missing-modality conditions for industrial inspection and structural health monitoring. A potential positive societal impact is

that more robust and lightweight crack segmentation systems may help reduce manual inspection burden, improve inspection efficiency, and support earlier detection of structural defects in resource-constrained deployment scenarios such as drones and edge devices.

At the same time, the proposed method may also introduce potential risks if it is over-relied upon in real-world safety-critical applications. In particular, incorrect predictions, such as missed crack regions or false alarms, may lead to inappropriate maintenance decisions if the system is used without human verification. In addition, although this work is developed for structural inspection, similar multimodal segmentation techniques could potentially be adapted to other monitoring scenarios.

To mitigate these risks, we emphasize that Prism is intended to serve as an assistive inspection tool rather than a fully autonomous decision-making system. In practical deployment, its predictions should be combined with human review and standard engineering inspection procedures. We also encourage future work to further study robustness, uncertainty calibration, and deployment safeguards in safety-critical environments.

N Compute Resources

All training experiments were conducted on a server with an Intel Xeon Platinum 8558 CPU and one NVIDIA L20 GPU, running Ubuntu 22.04.5. The GPU memory was 48 GB, and the system memory was 512 GB. Unless otherwise specified, each model was trained for 100 epochs with a batch size of 1. The average wall-clock training time for one complete run on a single dataset was approximately 4 hours. The total compute consumed for the main experiments in this paper was approximately 16 GPU-hours, excluding failed preliminary runs and hyperparameter tuning. For deployment experiments, inference was performed on an NVIDIA Jetson Orin NX Super device running Ubuntu 22.04.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main claims of the paper, including robustness under missing modalities, lightweight design, and improved segmentation performance. These claims are consistent with the methodology and the empirical results reported in the main paper and supplementary material.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The dedicated limitations paragraph identifies the opportunity to simplify the PaKD training pipeline, which currently uses a frozen teacher and structural objectives for implicit guidance, and outlines streamlined supervision and explicit texture generation as future directions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For each theoretical proposition, the supplementary material states the assumptions and provides derivations and proofs. The theory is presented as mechanism-level justification rather than as a claim of universal convergence.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main paper and appendix disclose the information needed to reproduce the main experiments, including the model architecture, AdamW optimizer and learning-rate schedule, frozen teacher configuration, evaluation metrics, stochastic missing-modality protocol, fixed dataset split ratio (7:1:2), batch size (1), and random seed (42). The implementation is publicly available at `https://github.com/Karl1109/Prism`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and usage instructions are publicly available at `https://github.com/Karl1109/Prism` to support reproduction of the main experiments. The datasets used in this work are existing public datasets introduced by prior work, and their sources are cited in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the main training and evaluation details, including datasets, optimizer, learning rate, training epochs, hardware platform, teacher model configuration, evaluation metrics, missing-modality protocol, dataset split ratio (7:1:2), batch size (1), and random seed (42).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The main tables report results from a single run with the random seed fixed to 42. They do not include error bars, confidence intervals, or formal statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper and supplementary material specify the main compute platform, including CPU, GPU, operating system, deployment hardware, and additional information on memory usage, training time, and approximate total compute budget required for the reported experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the research conducted in this paper conforms to the NeurIPS Code of Ethics and does not involve deceptive, harmful, or unethical experimental practices. The implementation is publicly available at `https://github.com/Karl1109/Prism`.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts, such as improved industrial inspection efficiency and safety, and potential negative impacts, such as over-reliance on automated predictions in safety-critical scenarios. It also outlines mitigation strategies, including human verification and cautious deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped dataset, or other asset with an obvious misuse profile that would require special safeguards for controlled release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper properly credits the original sources of the datasets, teacher model, and baseline methods used in this work. However, while the code repositories and data sources are cited, the licenses and terms of use of all existing datasets are not yet explicitly stated in the current submission.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and accompanying instructions are publicly available at `https://github.com/Karl1109/Prism`. The released assets are documented to support reproduction and appropriate use.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core methodology in this research. Any use of language models, if any, was limited to writing or editing assistance and does not affect the scientific content.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.