FM-ChangeNet: Learning Change through Pathwise Feature Transport

Roie Kazoom
Google Research
kazoomroie@google.com

George Leifman
Google Research
gleifman@google.com

Genady Beryozkin
Google Research
genady@google.com

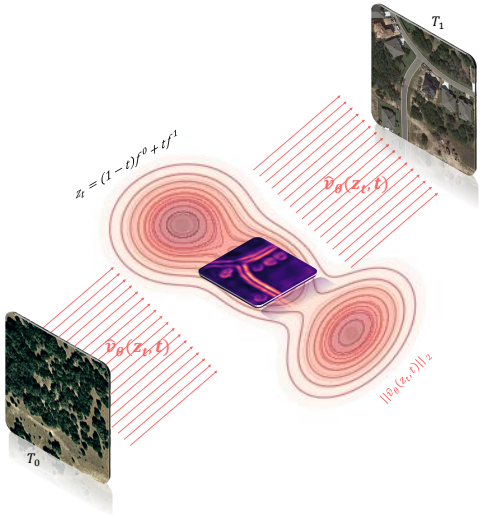


Figure 1: **Conceptual illustration of flow matching for change detection.** Given a bi-temporal image pair (T_0, T_1), the model defines an intermediate latent state $z_t = (1-t)f^0 + tf^1$ along a continuous feature-space interpolation between the encoded pre and post-temporal representations. A time-conditioned velocity field $\hat{v}_\theta(z_t, t)$ characterizes the local transport dynamics from T_0 toward T_1, while its magnitude $\|\hat{v}_\theta(z_t, t)\|_2$ provides a spatially localized cue for change.

Abstract

We present **FM-ChangeNet**, a *pathwise-supervised* framework for change detection that reformulates bi-temporal reasoning as *continuous transport in feature space* rather than static endpoint comparison. Given encoded pre and post-temporal representations, we construct intermediate latent states and learn a time-conditioned velocity field $\hat{v}_\theta(z_t, t)$ along the transformation trajectory. This pathwise formulation constrains the predictor over a continuum of intermediate states, providing a denser and less ambiguous supervision signal than conventional endpoint-only segmentation and enabling the model to capture temporal evolution explicitly. The learned velocity field is not only a transport mechanism but also an interpretable representation of change: its magnitude serves as a spatially localized change cue that helps distinguish true structural variation from nuisance effects such as illumination shifts and spatial misalignment. We develop a hierarchical multi-scale architecture with cross-temporal alignment, time-conditioned coarse-to-fine flow decoding, and a unified objective that couples flow supervision, trajectory consistency, spatial regularization, and segmentation loss. Experiments on remote sensing benchmarks show that the proposed framework produces more structured and robust change representations while achieving state-of-the-art performance.

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

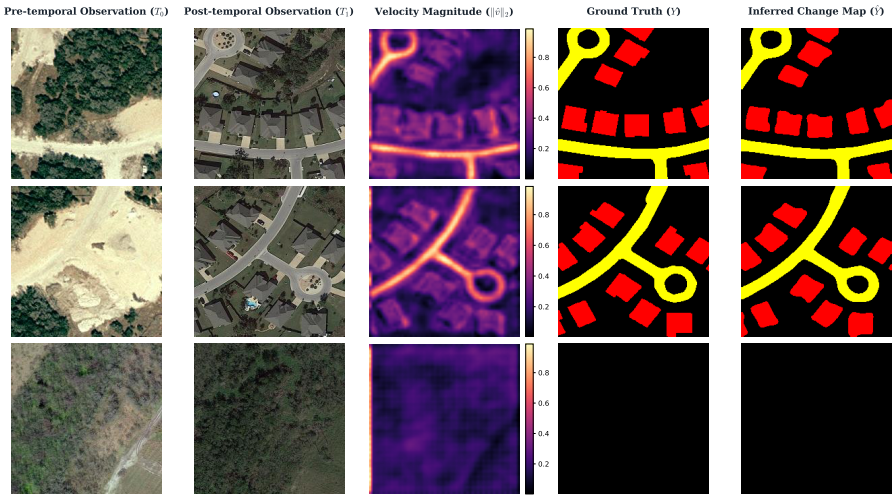


Figure 2: **Change detection as feature-space transport.** Given a bi-temporal pair (T_0, T_1), the encoder extracts feature representations $f^0 = E(T_0)$ and $f^1 = E(T_1)$. We model their relationship through a continuous latent path $z_t = (1-t)f^0 + tf^1$ and learn a time-conditioned velocity field $\hat{v}_\theta(z_t, t)$ using flow matching. The velocity magnitude $\|\hat{v}_\theta(z_t, t)\|_2$ provides a localized cue for change: unchanged regions require little transport, whereas structurally changed regions induce coherent high-magnitude responses. This transport-based representation aligns with the ground-truth mask Y and supports accurate prediction of the final change map $\hat{Y}$.

1 Introduction

Change detection (CD) aims to identify meaningful surface variations between bi-temporal images captured over the same geographical area at different times. It is a fundamental problem in remote sensing, with applications in urban expansion monitoring [3, 28], disaster assessment [36], and environmental conservation. Reliably isolating changes remains challenging due to nuisance factors such as illumination variation, seasonal effects, and viewpoint shifts, particularly in high-resolution imagery [8, 33]. Moreover, detecting subtle or small-scale changes often leads to high false-alarm rates when relying on local spectral differences [15]. Modern CD methods are dominated by Siamese architectures, where bi-temporal inputs are processed independently and fused via concatenation, differencing, or attention mechanisms. Early works such as FC-Siam [9] and IFNet [33] established this paradigm, while more recent approaches incorporate Transformers [4, 1] or dense multi-scale designs [11] to improve representation quality. Despite their strong empirical performance, these methods share a fundamental limitation: they treat the two observations as static snapshots and infer change from implicit feature comparisons. As a result, supervision is imposed primarily at the endpoint through the final change mask, while the underlying *temporal transition*, namely the process that transforms one state into the other, remains largely unconstrained. This limitation becomes especially pronounced under gradual structural evolution, cross-resolution discrepancies, and spatial misalignment, where endpoint differencing may confound true change with nuisance variation [5, 26]. Recent generative approaches attempt to address this limitation by modeling the distribution of change features, for example using diffusion models [2, 25]. While effective, such methods typically rely on stochastic sampling procedures and still do not explicitly characterize the transformation pathway between observations. Consequently, they improve the representation of change features, but do not directly supervise *how* pre-temporal structure evolves into post-temporal structure. This raises a fundamental question: **can change detection be formulated as learning the transformation itself, rather than detecting differences only at the endpoint?**

In this work, we answer this question by reframing change detection as a *pathwise-supervised feature-space transport problem*. Inspired by Flow Matching (FM) [17, 20, 27], we model the relationship between bi-temporal representations as a continuous transformation governed by a time-conditioned velocity field. Rather than relying solely on an endpoint-fused representation for mask prediction, FM-ChangeNet is constrained to predict transport over intermediate latent states. The fused deepest-scale

representation initializes cross-temporal alignment and coarse decoding, while the final prediction is conditioned on pathwise velocity estimates across the multi-scale trajectory. This reformulation changes the *supervision structure* of CD: instead of constraining the model only through the final segmentation output, we impose supervision along intermediate feature states. This provides a denser and more structured learning signal, encouraging temporally coherent transport dynamics rather than arbitrary endpoint features that merely fit the final mask. It also makes the learned velocity field an explicit change representation, where the magnitude of the predicted velocity acts as a spatially localized cue for structural variation. Fig. 1 illustrates our feature-space transport view of CD, and Fig. 2 shows how velocity magnitude localizes changed regions. Unlike conventional differencing-based approaches that directly use endpoint differences as the prediction signal, our model uses the endpoint displacement $f^1 - f^0$ as a supervised transport target over intermediate states z_t. Thus, the velocity magnitude is not an unsupervised heuristic, but a learned change-energy representation induced by pathwise supervision and refined through time-conditioned multi-scale decoding. To realize this formulation, we design a hierarchical multi-scale architecture that combines flow estimation with cross-temporal feature interaction [12, 31], enabling stable transformation learning under misalignment and noise [21]. The model integrates cross-temporal alignment, time-conditioned coarse-to-fine decoding, and a joint objective coupling flow supervision, trajectory consistency, spatial regularization, and segmentation, converting transport dynamics into dense change localization. **Contributions:**

We introduce a new *pathwise-supervised* formulation of change detection, in which bi-temporal change is modeled as feature-space transport and the predictor is constrained along a continuum of intermediate latent states rather than only through the final segmentation output.

We show, both theoretically and empirically, that pathwise supervision provides a stronger and less ambiguous learning signal than endpoint-only supervision, encouraging temporally coherent change representations instead of arbitrary feature differences that merely match the final mask.

We show that the learned velocity field serves as an explicit and interpretable representation of change, and instantiate this idea with a hierarchical multi-scale architecture that combines cross-temporal alignment, time-conditioned coarse-to-fine decoding, and joint flow and segmentation supervision for robust change localization under misalignment and noise.

2 Related Work

2.1 Change Detection in Remote Sensing

CD has evolved from early pixel-wise comparison methods to deep learning frameworks capable of modeling complex spatial and temporal variations [36]. Modern approaches are largely dominated by Siamese architectures, where bi-temporal images are processed independently and combined via feature differencing or fusion. Foundational works [9, 33] established this paradigm, while subsequent methods such as SNUNet-CD [11] and ICIF-Net [12] improved performance through multi-scale aggregation and cross-temporal interaction. With the introduction of Vision Transformers (ViTs) [10], recent approaches [4, 1, 31, 29] focus on modeling long-range dependencies and global context, achieving strong performance on large-scale datasets [3, 23]. Cross-resolution CD has been addressed by methods such as RACDNet [26], which explicitly handle discrepancies between heterogeneous sensors. Despite these advances, most existing methods share a common assumption: change is inferred from *static feature comparisons* between two observations. Whether through differencing, concatenation, or attention-based fusion, the temporal relationship between the two states is not explicitly modeled. This limitation becomes pronounced in scenarios involving gradual structural evolution, cross-resolution misalignment, or subtle changes. Generative approaches attempt to alleviate these issues by modeling the distribution of change features. GAN-based methods [30] and diffusion-based models [2, 25] improve robustness to noise and domain shifts, but typically rely on stochastic sampling procedures and do not explicitly capture the transformation pathway between observations. This suggest that existing methods implicitly treat CD as a static comparison problem, motivating formulations that explicitly model temporal evolution.

2.2 Flow Matching and Feature Transformation

Flow Matching (FM) has recently emerged as a deterministic alternative to diffusion-based generative modeling [17, 20]. Instead of learning a stochastic denoising process, FM directly parameterizes

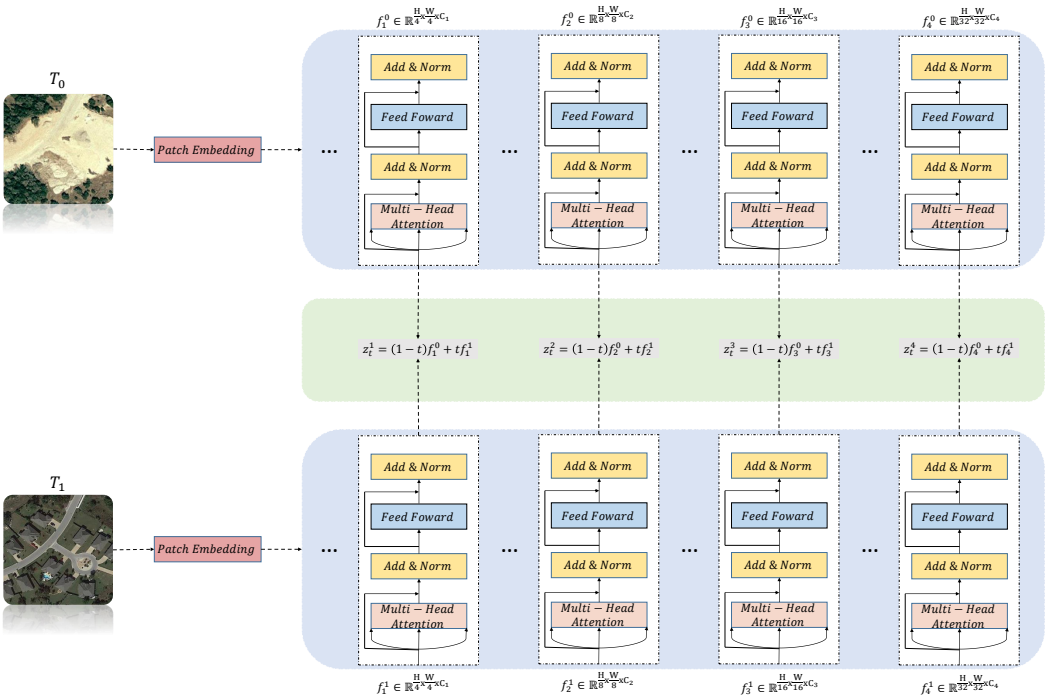


Figure 3: **Encoder and temporal feature construction in the proposed flow-matching framework for CD.** Given a bi-temporal image pair (T_0, T_1), a shared ViT encoder extracts hierarchical multi-scale feature maps $\{f_1^0, f_2^0, f_3^0, f_4^0\}$ and $\{f_1^1, f_2^1, f_3^1, f_4^1\}$ from the two observations. At each scale, a time-conditioned interpolated representation is constructed as $z_t^i = (1-t)f_i^0 + tf_i^1$, which models the temporal evolution between the two feature states in latent space. These interpolated multi-scale representations passed to the subsequent alignment and decoding stages shown in Fig. 4.

a time-dependent vector field $v_\theta(x,t)$ that transports samples along a continuous path between source and target distributions via a probability flow ODE [27]. Its standard objective is $\mathcal{L}_{\text{FM}} = \mathbb{E}_{t\sim\mathcal{U}(0,1),x_t}\left[\|v_\theta(x_t,t) - v^\star(x_t,t)\|^2\right]$, where $v^\star$ denotes the target velocity field induced by the chosen interpolation path. This formulation enables efficient and stable learning of structured transformations without iterative sampling. Existing FM methods are mainly designed for generative modeling, where the goal is to transport one distribution into another or synthesize samples from a learned flow. In contrast, CD does not require generating a target observation, but identifying *where* meaningful temporal variation occurs between two observations. We therefore adapt FM as a change-specific supervision mechanism: the learned velocity field is defined over intermediate bi-temporal feature states and provides pathwise constraints beyond endpoint-only prediction. Its magnitude further acts as a spatially resolved cue for change, making FM suitable for dense localization under nuisance appearance variation and misalignment.

3 Methodology

3.1 Problem Formulation

Given a bi-temporal pair (T_0, T_1), with $T_0, T_1 \in \mathbb{R}^{H\times W\times 3}$, the goal is to predict a dense change map: $Y \in \{0,1,\ldots,C-1\}^{H\times W}$, where C denotes the number of classes. We interpret CD as learning a *transformation operator* that maps the representation of T_0 to that of T_1. Let $\mathcal{E}$ denote a feature encoder, and define $f^0 = \mathcal{E}(T_0)$ and $f^1 = \mathcal{E}(T_1)$. Rather than inferring change from static feature comparisons, we model the relationship between f^0 and f^1 as a continuous trajectory $\{z_t\}_{t\in[0,1]}$ governed by a time-dependent velocity field v_θ: $\frac{dz_t}{dt} = v_\theta(z_t,t), \quad z_0 = f^0,\ z_1 = f^1$. CD reduces to identifying regions where the induced transformation exhibits high magnitude. We refer to this formulation as *feature-space transport for CD*. Figs. 3-4 summarize the full pipeline, from

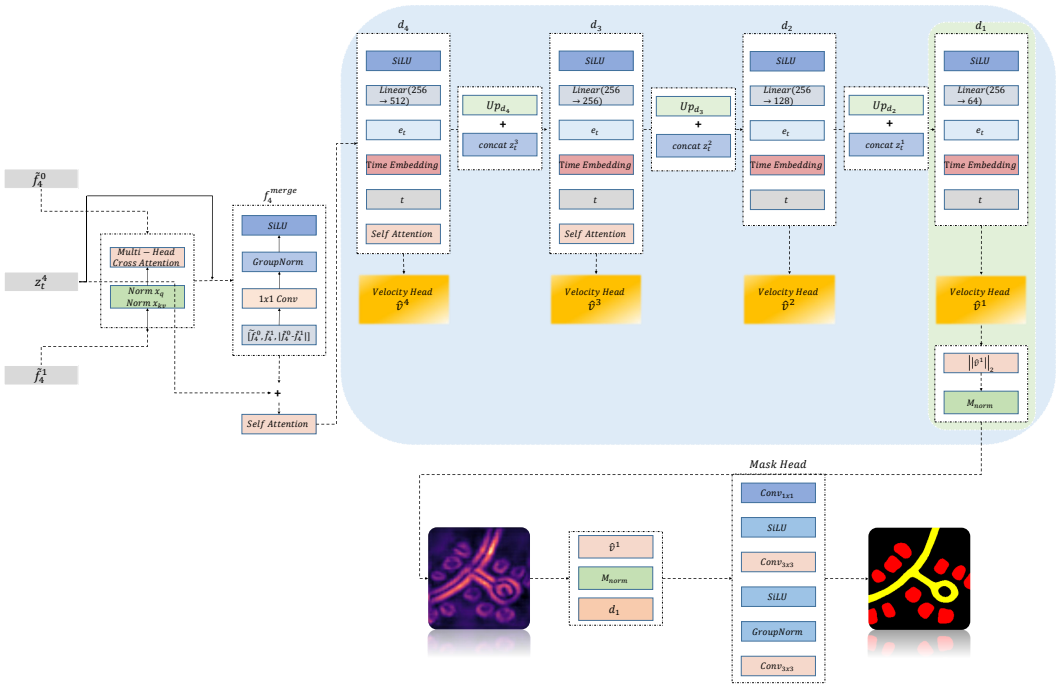


Figure 4: **Continuation of Fig. 3: alignment, coarse-to-fine decoding, and change prediction.** Taking the interpolated multi-scale latent features $\{z_t^1, z_t^2, ..., z_t^n\}$ from Fig. 3 as input, the deepest representation is first refined through symmetric cross-attention between the bi-temporal features and fused by a token-merging module to obtain a shared coarse feature initialization. A time-conditioned coarse-to-fine decoder then progressively refines the representations across scales and predicts a velocity field at each decoding stage. The normalized finest-scale velocity magnitude is combined with the last decoder feature through a segmentation head to generate the final change map.

multi-scale feature extraction and temporal interpolation to alignment, time-conditioned decoding, and velocity-based mask prediction.

3.2 Multi-Scale Feature Extraction

We employ a Vision Transformer (ViT) encoder $\mathcal{E}$ to extract hierarchical feature representations: $\{f_1, f_2, .., f_n\} = \mathcal{E}(T)$, where $f_s \in \mathbb{R}^{C_s \times H_s \times W_s}$ corresponds to scale s. For the bi-temporal pair (T_0, T_1), we obtain: $\{f_s^0\}, \quad \{f_s^1\}, \quad s \in \{1, 2, .., n\}$.

3.3 Flow Matching in Feature Space

We model the transformation between feature representations as a continuous flow. For each scale s, we define an interpolation: $z_t^{(s)} = (1-t)f_s^0 + tf_s^1, \quad t \in [0, 1]$, where $t \sim \mathcal{U}(0, 1)$ during training. The objective is to learn a velocity field: $\hat{v}_\theta^{(s)} = \hat{v}_\theta(z_t^{(s)}, t)$, that approximates the instantaneous displacement along the transformation trajectory. Under linear interpolation, the optimal velocity field is: $v^{(s)} = f_s^1 - f_s^0$, corresponding to a constant transport direction between endpoints. Learning $\hat{v}_\theta$ can therefore be interpreted as estimating the underlying *transport field* that aligns source and target representations. Unlike diffusion models, which rely on stochastic sampling, this formulation directly models a deterministic transformation in feature space. This formulation is not equivalent to raw feature differencing. Conventional differencing compares only the endpoints through quantities such as $|f_s^1 - f_s^0|$, and therefore supervises the model only through a single static comparison. In contrast, our predictor is conditioned on both the intermediate latent state $z_t^{(s)}$ and the temporal position t, and is trained over a continuum of sampled states along the path between f_s^0 and f_s^1. This imposes consistency across a family of intermediate representations rather than only at the endpoint, yielding

a more structured and constrained predictor of temporal change. We adopt linear interpolation as the default transport path because it defines a simple canonical target field and yields stable optimization. To verify that the proposed framework is not tied to this specific choice, we evaluate robustness to nonlinear interpolation, learned interpolation, and alternative time-sampling strategies in Sec. B. Although the canonical target velocity under linear interpolation is the constant field $f_1 - f_0$, the learning problem is not trivial. The predictor is not trained to output a single static endpoint difference in isolation; instead, it is conditioned on the intermediate latent state z_t, the temporal position t, and the aligned multi-scale feature context produced by the encoder and decoder. The benefit therefore does not come from making the target field itself more complex, but from constraining the predictor to remain consistent across a continuum of intermediate states along the transport path.

3.4 Cross-Temporal Alignment via Attention

Accurate transport prediction benefits from cross-temporal alignment context, but the canonical flow target is still defined between the encoder features f_s^0 and f_s^1. We introduce a symmetric cross-attention mechanism at the deepest scale: $\tilde{f}_n^0 = \text{Attn}(f_n^0, f_n^1), \quad \tilde{f}_n^1 = \text{Attn}(f_n^1, f_n^0)$. This operation establishes correspondences between temporal features, reducing ambiguity due to misregistration or structural variation. This step provides an aligned contextual representation that conditions the decoder, helping the predicted velocity distinguish semantic change from nuisance displacement. Without explicit cross-temporal alignment, the estimated velocity field may partially reflect spatial mismatch rather than true evolution. By refining the deepest features through symmetric attention, the model is encouraged to resolve long-range correspondences before transport estimation, making the subsequent flow prediction more faithful to change rather than nuisance displacement. We then construct a fused representation: $f_n^{\text{merge}} = \phi\left([\tilde{f}_n^0, \tilde{f}_n^1, |\tilde{f}_n^0 - \tilde{f}_n^1|]\right)$, where ϕ is a learnable projection.

3.5 Time-Conditioned Multi-Scale Decoder

We design a decoder that estimates the transformation in a coarse-to-fine manner while conditioning on a temporal embedding. The scalar value t is mapped to $e_t = \mathcal{T}(t) \in \mathbb{R}^{256}$ using a two-layer MLP with SiLU activation: $e_t = W_2\,\text{SiLU}(W_1 t + b_1) + b_2$. At the deepest scale, we initialize $d_n = \mathcal{F}_n\left(z_t^{(n)} + f_n^{\text{merge}}, e_t\right)$, where $z_t^{(n)}$ carries the canonical transport path and f_n^{merge} provides aligned cross-temporal context. For finer scales, we recursively compute $d_{n-s} = \mathcal{F}_{n-s}\left(\text{Up}(d_{n-s+1}), z_t^{(n-s)}, e_t\right)$, for $s \in \{1, \ldots, n-1\}$. At each scale, a velocity head predicts $\hat{v}^{(s)} = \mathcal{H}_s(d_s)$. This design captures both global structural transformations and localized boundary changes, progressively translating transport dynamics into dense change evidence.

3.6 Velocity-Based Change Representation

At the finest scale, we derive a change representation from the predicted velocity field by computing the channel-wise magnitude $M = \|\hat{v}^{(1)}\|_2$, which measures local transformation intensity. Under the transport formulation, changed regions are expected to induce larger feature-space displacement, while unchanged regions should remain near-stationary with $v \approx 0$. To avoid amplifying low-energy noise in nearly unchanged image pairs, we do not use per-image min-max normalization. Instead, we normalize M using fixed percentile statistics estimated on the validation set: $M_{\text{norm}} = \text{clip}\left(\frac{M - q_1}{q_{99} - q_1 + \epsilon}, 0, 1\right)$, where q_1 and q_{99} denote the validation-set 1st and 99th percentiles of velocity magnitudes. The normalized magnitude is then combined with directional velocity information and decoder features to produce segmentation logits: $\hat{Y}_{\text{logits}} = \mathcal{G}\left([\hat{v}^{(1)}, M_{\text{norm}}, d_1]\right)$.

3.7 Training Objective

The training objective combines transport estimation, trajectory consistency, spatial regularization, and segmentation supervision. These terms are designed not only to fit a velocity field, but to make the learned transport dynamics semantically meaningful for dense change localization. The objective couples pathwise supervision in feature space with endpoint supervision, ensuring that the predicted transport remains both temporally coherent and predictive of the final change mask.

Flow Matching Loss $\mathcal{L}_{\text{flow}} = \sum_{s=1}^{4} w_s \|\hat{v}^{(s)} - v^{(s)}\|_2^2$, which encourages the learned velocity field to approximate the underlying transport field. Unlike endpoint-only supervision, this term constrains the model along sampled intermediate states of the temporal trajectory, encouraging the predictor to learn coherent transport dynamics rather arbitrary feature differences that only explain the final label.

Consistency Loss To enforce coherence along the trajectory, we reconstruct endpoint features: $\hat{f}_1^1 = z_1^{(1)} + (1-t)\hat{v}^{(1)}, \quad \hat{f}_1^0 = z_1^{(1)} - t\hat{v}^{(1)}, \mathcal{L}_{\text{cons}} = \|\hat{f}_1^1 - f_1^1\|_2^2 + \|\hat{f}_1^0 - f_1^0\|_2^2$. This enforces bidirectional consistency of the learned transformation. It ties local transport predictions back to the endpoint representations, discouraging solutions that appear locally plausible along the path but fail to preserve the relationship between the pre and post-temporal features.

Total Variation Regularization $\mathcal{L}_{\text{tv}} = \|\nabla_h \hat{v}^{(1)}\|_1 + \|\nabla_w \hat{v}^{(1)}\|_1$, which promotes spatial smoothness in the velocity field. This regularization suppresses noisy or fragmented transport estimates and encourages spatially coherent change dynamics.

Segmentation Loss $\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{Dice}}$. While flow supervision learns the transformation dynamics, segmentation supervision anchors these dynamics to the target change labels. This term converts transport-based cues into dense pixel-level predictions and ensures that the learned velocity representation remains directly useful.

Final Objective $\mathcal{L} = \lambda_{\text{flow}}\mathcal{L}_{\text{flow}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}} + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}$.

3.8 Transport-Based Change Principles

We now provide a theoretical justification for the proposed formulation. Our central claim is not only that change can be interpreted as transport in feature space, but that this view induces a more structured supervision mechanism than conventional endpoint-only change prediction. In particular, by learning a velocity field over intermediate latent states, the model is constrained to explain how pre-temporal features evolve into post-temporal features along an entire trajectory rather than only through the final segmentation output. For clarity, we present the analysis at a single scale and omit the superscript (s) unless needed.

Theorem 1 (Canonical Transport Field). Let $f^0, f^1 \in \mathbb{R}^{C\times H\times W}$ denote the encoded features of the pre- and post-temporal images, and define the interpolation path

$$z_t = (1-t)f^0 + tf^1, \qquad t \in [0,1]. \tag{1}$$

Then z_t is generated by the constant velocity field $v^\star(z_t, t) = f^1 - f^0$, that is,

$$\frac{dz_t}{dt} = v^\star(z_t, t), \qquad z_0 = f^0,\ z_1 = f^1. \tag{2}$$

This identifies the canonical transport target induced by the chosen interpolation path and provides the reference field used for supervision.

Proposition 1 (Optimality of the Flow Matching Target). Consider the pathwise flow-matching objective

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\left[\left\|\hat{v}_\theta(z_t,t) - (f^1 - f^0)\right\|_2^2\right]. \tag{3}$$

Its unique minimizer in L^2 is the canonical transport field $v^\star(z_t,t) = f^1 - f^0$ almost everywhere along the path. Thus, under the chosen trajectory, flow matching does not merely compare endpoints, but supervises the predictor to remain consistent with the same transport field across a continuum of intermediate states.

Proposition 2 (Change-Energy Principle). Assume that, after cross-temporal alignment, the feature difference at spatial location p can be decomposed as $f_p^1 - f_p^0 = s_p + r_p$, where s_p denotes the semantic change component and r_p denotes residual nuisance variation due to imperfect alignment, illumination, or noise. If $\|r_p\|_2 \le \delta_p$, then the canonical transport magnitude satisfies

$$\|s_p\|_2 - \delta_p \le \|v_p^\star\|_2 \le \|s_p\|_2 + \delta_p \tag{4}$$

Thus, when the content is unchanged and the residual nuisance is small, $\|v_p^\star\|_2$ remains close to zero. Conversely, when the semantic change magnitude $\|s_p\|_2$ is large relative to δ_p, the transport

magnitude is lower-bounded and therefore expected to be large. This motivates using velocity magnitude as a localized change-energy cue: semantically stable regions require little transport, while structurally changed regions induce larger feature-space displacement.

Proposition 3 (Pathwise Supervision Advantage). Endpoint-only objectives supervise the predictor only through the final change map $\hat{Y}$, whereas flow matching constrains the model on the entire trajectory $\{z_t\}_{t\in[0,1]}$. Consequently, the proposed objective enforces transport consistency over a continuum of intermediate states rather than only at the endpoint. Proposition 3 highlights the main conceptual advantage of the proposed formulation. In endpoint-only CD, many latent transformations may produce similar final masks, leaving the temporal reasoning process underconstrained. Pathwise supervision restricts the feasible solution space by requiring the predictor to remain consistent across intermediate latent states sampled along the transport path. As a result, the model is encouraged to learn temporally coherent change representations rather than arbitrary endpoint feature discrepancies.

Proposition 4 (Additional Representation-Level Constraints). Let θ parameterize the full prediction pipeline, including the velocity predictor $\hat{v}_\theta$ and the final change predictor $\hat{Y}_\theta$. Define

$$\mathcal{S}_{\text{end}} = \left\{\theta : \ell_{\text{mask}}(\hat{Y}_\theta, Y) = 0\right\}, \tag{5}$$

and

$$\mathcal{S}_{\text{path}} = \left\{\theta : \ell_{\text{mask}}(\hat{Y}_\theta, Y) = 0 \ \text{ and } \ \hat{v}_\theta(z_t, t) = v^\star(z_t, t) \ \text{ for a.e. } \ t \in [0,1]\right\}. \tag{6}$$

Then,

$$\mathcal{S}_{\text{path}} \subseteq \mathcal{S}_{\text{end}}. \tag{7}$$

Moreover, if there exists an endpoint-correct parameterization whose internal transport field disagrees with $v^\star$ on a set of nonzero measure, then the inclusion is strict:

$$\mathcal{S}_{\text{path}} \subset \mathcal{S}_{\text{end}}. \tag{8}$$

Thus, pathwise supervision imposes representation-level constraints that are absent from endpoint-only supervision, selecting the subset of endpoint-correct solutions whose internal transport representation is consistent with the canonical feature-space transition. These results formalize the role of flow matching in our framework: Theorem 1 and Proposition 1 define the canonical transport target, Proposition 2 motivates velocity magnitude as a localized change cue under aligned features, and Propositions 3-4 show that pathwise supervision imposes stability and transport-consistency constraints beyond endpoint-only prediction. These principles motivate the appendix studies on velocity-only prediction, matched-capacity endpoint baselines, and robustness to nuisance variation.

4 Results

We evaluate the proposed framework on three standard remote sensing CD benchmarks: LEVIR-CD [3], WHU-CD [14], and DSIFN-CD [33]. LEVIR-CD [3] focuses on small, densely distributed building changes in high-resolution aerial imagery, WHU-CD [14] contains larger urban scenes with stronger viewpoint and illumination variation, and DSIFN-CD [33] is more diverse and challenging, with richer clutter and larger intra-class variation. We follow the official splits, and apply flipping, rotation, and cropping during training. We report Precision, Recall, F1, IoU, and overall accuracy (OA), with F1 and IoU as the primary metrics due to the strong foreground-background imbalance typical of CD. Table 1 compares our FM framework against representative CNN-based, Transformer-based, and Mamba-based methods on LEVIR-CD, WHU-CD, and DSIFN-CD. Our method achieves the best F1, IoU, and OA on LEVIR-CD, and the best Precision, Recall, and OA on WHU-CD.

Table 2 provides a controlled test of the core assumption behind FM-ChangeNet: modeling change as a learned transport field from f^0 to f^1 is more effective than relying on static endpoint comparisons. Importantly, the linear path does not make the formulation equivalent to simple differencing. While the canonical target field is $f^1 - f^0$, the model is trained to predict this field from many intermediate states $z_t = (1-t)f^0 + tf^1$ under time conditioning and multi-scale aligned context. This turns a single endpoint displacement into a dense pathwise supervision signal, forcing the predictor to remain consistent along the entire feature trajectory. All variants use matched capacity whenever applicable, including the same encoder, alignment module, decoder capacity, and prediction head, while changing only the temporal modeling mechanism. The full model consistently outperforms static differencing, removing time conditioning, direct fused-feature prediction, endpoint feature

Table 1: Quantitative comparison with SOTA methods on LEVIR-CD, WHU-CD, and DSIFN-CD. Values are reported as percentages (%). "–" indicates that the corresponding was not reported.

Type	Models	LEVIR-CD					WHU-CD					DSIFN-CD				
		Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
CNN-based	FC-EF [9]	86.91	80.79	83.74	72.02	98.39	86.63	81.63	84.05	72.50	98.33	72.61	52.73	61.09	43.98	88.59
	FC-Siam-Diff [9]	89.53	82.87	86.07	75.55	98.67	85.43	86.31	85.87	75.24	98.51	59.67	65.71	62.54	45.50	86.63
	FC-Siam-Conc [9]	91.14	83.75	87.29	77.45	98.77	86.41	85.34	85.87	75.24	98.51	66.45	54.21	59.71	42.56	87.57
	IFNet [33]	94.02	82.93	88.13	78.77	98.87	95.34	89.70	92.43	85.93	99.21	67.86	53.94	60.10	42.96	87.83
	SNUNet [11]	89.42	87.17	88.28	79.02	98.81	94.13	91.22	92.65	86.31	99.22	60.60	72.89	66.18	49.45	87.34
Transformer-based	SwinUnet [32]	91.24	87.10	89.12	80.37	98.89	92.81	93.42	93.11	87.12	99.27	–	–	–	–	–
	BIT [4]	89.24	89.37	89.31	80.68	98.92	93.30	92.65	92.97	86.87	99.25	68.36	70.18	69.26	52.97	89.41
	ChangeFormer [1]	92.42	88.64	90.49	82.63	99.04	92.49	94.69	93.58	87.93	99.30	88.48	84.94	86.67	76.48	95.56
	MSCANet [35]	92.45	88.85	90.62	82.84	99.05	95.14	92.90	94.01	88.69	99.37	–	–	–	–	–
	Paformer [18]	92.51	89.33	90.89	83.31	99.08	94.55	93.91	94.23	89.09	99.39	–	–	–	–	–
	DARNet [16]	92.34	89.51	90.90	83.32	99.08	94.67	94.12	94.39	89.38	99.41	–	–	–	–	–
	ACABFNet [24]	92.54	89.47	90.98	83.45	99.09	94.61	94.03	94.32	89.24	99.40	–	–	–	–	–
	NeXt2Former-CD [29]	–	–	92.10	85.40	99.20	–	–	95.50	91.40	99.65	–	–	–	–	–
Mamba	RS-Mamba [7]	91.56	90.64	91.10	83.65	99.10	94.94	94.61	94.77	90.07	99.44	–	–	–	–	–
	ChangeMamba [6]	92.33	91.22	91.77	84.79	99.17	95.02	94.98	95.00	90.48	99.46	–	–	–	–	–
	CDMamba [34]	92.41	92.05	92.23	85.58	99.21	95.31	95.17	95.24	90.91	99.49	–	–	–	–	–
Diffusion-based	DDPM-CD [2]	–	–	90.90	83.30	99.10	–	–	92.70	86.30	99.40	–	–	–	–	–
FM	**Ours**	**92.70**	**92.20**	**92.45**	**85.96**	**99.24**	**95.39**	**95.49**	**95.44**	**91.28**	**99.66**	**90.02**	**87.93**	**88.96**	**80.12**	**95.61**

regression, and final-mask-only supervision across all datasets. These results show that the gains are not due to architecture capacity alone, but to the linear transport formulation itself: it provides a simple, stable, and well-defined target field while still allowing the network to learn nonlinear, spatially structured change evidence through the conditioned decoder and segmentation head.

Table 2: **Matched-capacity comparison against static and endpoint-supervised alternatives.** Variants use the same encoder, alignment module, decoder capacity, and prediction head whenever applicable. The comparison isolates whether the gains arise from explicit pathwise transport modeling rather than architecture capacity alone. The endpoint feature-regression baseline directly predicts the endpoint displacement $f^1 - f^0$ without sampled intermediate states z_t or time conditioning t.

Dataset	Method	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
LEVIR-CD	FM-ChangeNet	92.70	92.20	92.45	85.96	99.24
	Static difference, same encoder-decoder	84.94	83.69	84.31	72.88	98.37
	No time conditioning, same decoder	85.62	84.52	85.07	74.02	98.46
	Endpoint feature regression, no pathwise z_t, t	76.88	75.36	76.11	61.44	97.21
	Direct prediction from fused features	86.71	85.66	86.18	75.72	98.58
	Final-mask supervision only	88.04	87.22	87.63	77.98	98.73
WHU-CD	FM-ChangeNet	95.39	95.49	95.44	91.28	99.66
	Static difference, same encoder-decoder	88.46	87.37	87.91	78.43	98.71
	No time conditioning, same decoder	89.18	88.10	88.64	79.60	98.79
	Endpoint feature regression, no pathwise z_t, t	80.14	78.62	79.37	65.80	96.89
	Direct prediction from fused features	90.09	89.07	89.58	81.13	98.87
	Final-mask supervision only	91.71	90.81	91.26	83.92	99.04
DSIFN-CD	FM-ChangeNet	90.02	87.93	88.96	80.12	95.61
	Static difference, same encoder-decoder	81.79	80.50	81.14	68.27	94.31
	No time conditioning, same decoder	82.49	81.24	81.86	69.29	94.46
	Endpoint feature regression, no pathwise z_t, t	73.45	70.98	72.19	56.49	91.22
	Direct prediction from fused features	83.31	82.17	82.73	70.55	94.58
	Final-mask supervision only	84.96	84.01	84.48	73.13	94.87

5 Conclusion and Future Work

We reformulated remote sensing CD as a pathwise-supervised transport problem in feature space. Instead of relying on static endpoint comparisons, the proposed method learns a time-conditioned velocity field that models how pre-temporal features evolve into post-temporal features through intermediate states. This yields a more structured supervision signal than endpoint-only segmentation and produces an explicit change representation whose magnitude serves as a localized indicator of variation. Combined with cross-temporal alignment and hierarchical multi-scale decoding, the framework provides a principled alternative to conventional differencing-based methods and improves localization under noise, misalignment, and subtle structural change. A current limitation is that the formulation remains deterministic and has not yet been evaluated under stronger distribution shifts, severe temporal misregistration, or large-scale efficiency constraints. **Future work** includes extending the framework to multi-temporal sequences, incorporating uncertainty estimation, and integrating additional modalities such as SAR or textual metadata.

References

[1] Wele Gedara Chaminda Bandara and Vishal M Patel. A transformer-based siamese network for change detection. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210. IEEE, 2022.

[2] Wele Gedara Chaminda Bandara, Nithin Gopalakrishnan Nair, and Vishal Patel. Ddpm-cd: Denoising diffusion probabilistic models as feature extractors for remote sensing change detection. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 5250–5262, 2025.

[3] Hao Chen and Zhenwei Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote sensing*, 12(10):1662, 2020.

[4] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 2021. doi: 10.1109/TGRS.2021.3095166. URL `https://doi.org/10.1109/TGRS.2021.3095166`.

[5] Hao Chen, Haotian Zhang, Keyan Chen, Chenyao Zhou, Song Chen, Zhengxia Zou, and Zhenwei Shi. Continuous cross-resolution remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20, 2023. doi: 10.1109/TGRS.2023.3325829.

[6] Hongruixuan Chen, Jian Song, Chengxi Han, Junshi Xia, and Naoto Yokoya. Changemamba: Remote sensing change detection with spatiotemporal state space model. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024.

[7] Keyan Chen, Bowen Chen, Chenyang Liu, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. Rsmamba: Remote sensing image classification with state space model. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024.

[8] Yongqi Chen, Shou Feng, Chunhui Zhao, Yingjie Tang, Junjie Tang, Di Dong, and Nan Su. Deep learning for high-resolution remote sensing change detection: Task taxonomy, datasets, and perspectives. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP(99):1–19, 2025. doi: 10.1109/JSTARS.2025.3622204.

[9] Rodrigo Caye Daudt, Bertrand Le Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. In *2018 25th IEEE international conference on image processing (ICIP)*, pages 4063–4067. IEEE, 2018.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020.

[11] Sheng Fang, Kaiyu Li, Jinyuan Shao, and Zhe Li. Snunet-cd: A densely connected siamese network for change detection of vhr images. *IEEE Geoscience and Remote Sensing Letters*, 2021. doi: 10.1109/LGRS.2021.3056416. URL `https://doi.org/10.1109/LGRS.2021.3056416`.

[12] Yuchao Feng, Honghui Xu, Jiawei Jiang, Hao Liu, and Jianwei Zheng. Icif-net: Intra-scale cross-interaction and inter-scale feature fusion network for bitemporal remote sensing images change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. doi: 10.1109/TGRS.2022.3168331. URL `https://ieeexplore.ieee.org/document/9759285`.

[13] Yuchao Feng, Jiawei Jiang, Honghui Xu, and Jianwei Zheng. Change detection on remote sensing images using dual-branch multi-level inter-temporal network. *IEEE Transactions on Geoscience and Remote Sensing*, 2023. doi: 10.1109/TGRS.2023.3241257. URL `https://doi.org/10.1109/TGRS.2023.3241257`.

[14] Shunping Ji, Shiqing Wei, and Meng Lu. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions on geoscience and remote sensing*, 57(1):574–586, 2018.

[15] Tao Lei, Xinzhe Geng, Hailong Ning, Zhiyong Lv, Maoguo Gong, Yaochu Jin, and Asoke K Nandi. Ultralightweight spatial–spectral feature cooperation network for change detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–14, 2023.

[16] Ziming Li, Chenxi Yan, Ying Sun, and Qinchuan Xin. A densely attentive refinement network for change detection based on very-high-resolution bitemporal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2022.

[17] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[18] Mengxi Liu, Qian Shi, Zhuoqun Chai, and Jianlong Li. Pa-former: Learning prior-aware transformer for remote sensing building change detection. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[19] Mengxi Liu, Qian Shi, Andrea Marinoni, Da He, Xiaoping Liu, and Liangpei Zhang. Super-resolution-based change detection network with stacked attention module for images with different resolutions. *IEEE Transactions on Geoscience and Remote Sensing*, 2022. doi: 10.1109/TGRS.2021.3091758. URL `https://doi.org/10.1109/TGRS.2021.3091758`.

[20] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[21] Seyedehanita Madani, Rama Chellappa, and Vishal M. Patel. Diffregcd: Integrated registration and change detection with diffusion features. *arXiv preprint arXiv:2511.07935*, 2025. URL `https://arxiv.org/abs/2511.07935`.

[22] Ruizhe Shao, Chun Du, Hao Chen, and Jun Li. Sunet: Change detection for heterogeneous remote sensing images from satellite and uav using a dual-channel fully convolution network. *Remote Sensing*, 13(18):3750, 2021. doi: 10.3390/rs13183750. URL `https://www.mdpi.com/2072-4292/13/18/3750`.

[23] Li Shen, Yao Lu, Hao Chen, Hao Wei, Donghai Xie, Jiabao Yue, Rui Chen, Shouye Lv, and Bitao Jiang. S2looking: A satellite side-looking dataset for building change detection. *Remote Sensing*, 13(24):5094, 2021.

[24] Lei Song, Min Xia, Liguo Weng, Haifeng Lin, Ming Qian, and Binyu Chen. Axial cross attention meets cnn: Bibranch fusion network for change detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:21–32, 2022.

[25] Kai Tang and Jin Chen. Changeanywhere: Sample generation for remote sensing change detection via semantic latent diffusion model. *arXiv preprint arXiv:2404.08892*, 2024.

[26] Juan Tian, Daifeng Peng, Haiyan Guan, Lidong Wu, Hongfeng Li, and Maoguo Gong. RACD-Net: Resolution- and alignment-aware change detection network for optical remote sensing imagery. *Remote Sensing*, 14(18):4527, 2022. doi: 10.3390/rs14184527.

[27] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

[28] Adam Van Etten, Daniel Hogan, Jesus Martinez Manso, Jacob Shermeyer, Nicholas Weir, and Ryan Lewis. The multi-temporal urban development spacenet dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2021.

[29] Yufan Wang, Sokratis Makrogiannis, and Chandra Kambhamettu. Next2former-cd: Efficient remote sensing change detection with modern vision architectures. *arXiv preprint arXiv:2602.18717*, 2026.

[30] Chen Wu, Bo Du, and Liangpei Zhang. Fully convolutional change detection framework with generative adversarial network for unsupervised, weakly supervised and regional supervised change detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9774–9788, 2023. doi: 10.1109/TPAMI.2023.3237896.

[31] Qingsong Xu, Yilei Shi, Jianhua Guo, Chaojun Ouyang, and Xiao Xiang Zhu. UCDFormer: Unsupervised change detection using a transformer-driven image translation. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–17, 2023. doi: 10.1109/TGRS.2023.3305334.

[32] Chengyao Zhang, Lei Wang, Shiyong Cheng, and Yang Li. Swinsunet: Pure transformer network for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[33] Chenxiao Zhang, Peng Yue, Deodato Tapete, Liangcun Jiang, Boyi Shangguan, Li Huang, and Guangchao Liu. A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:183–200, 2020.

[34] Haotian Zhang, Keyan Chen, Chenyang Liu, Hao Chen, Zhengxia Zou, and Zhenwei Shi. Cdmamba: Incorporating local clues into mamba for remote sensing image binary change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[35] Xingjian Zheng, Xin Lin, Linbo Qing, and Xianfeng Ou. Semantic-aware remote sensing change detection with multi-scale cross-attention. *Sensors*, 25(9):2813, 2025.

[36] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE geoscience and remote sensing magazine*, 5(4):8–36, 2017.

A Proofs for Transport-Based Change Principles

For simplicity, we present the analysis at a single scale and omit the scale index. The goal of this appendix is to formalize four properties of the proposed formulation: (i) the canonical transport target induced by the chosen interpolation path, (ii) the optimality of the corresponding flow-matching objective along this path, (iii) the conditions under which velocity magnitude can serve as a localized feature-space change cue, and (iv) the additional constraints imposed by supervising intermediate latent states rather than only the final prediction. Importantly, the analysis focuses on a constructed feature-space trajectory that provides dense supervision between the two observed endpoints, rather than requiring intermediate temporal observations.

A.1 Setup

Let $f^0, f^1 \in \mathbb{R}^{C\times H\times W}$ denote the encoded pre- and post-temporal feature tensors. We define the interpolation path

$$z_t = (1-t)f^0 + tf^1, \qquad t \in [0,1]. \tag{9}$$

The learned model predicts a time-dependent velocity field $\hat{v}_\theta(z_t, t)$ and is trained to match the displacement induced by this chosen path. For the linear path above, this target displacement is $f^1 - f^0$.

A.2 Proof of Theorem 1

Theorem 1 (Canonical Transport Field). Let

$$z_t = (1-t)f^0 + tf^1. \tag{10}$$

Then the path z_t is generated by the constant velocity field

$$v^\star(z_t, t) = f^1 - f^0. \tag{11}$$

Proof. Differentiating z_t with respect to t gives

$$\frac{dz_t}{dt} = \frac{d}{dt}\left((1-t)f^0 + tf^1\right) = -f^0 + f^1 = f^1 - f^0. \tag{12}$$

Hence

$$\frac{dz_t}{dt} = v^\star(z_t, t), \tag{13}$$

with $z_0 = f^0$ and $z_1 = f^1$. Therefore, the chosen interpolation path is exactly generated by the constant field $v^\star$. □

A.3 Proof of Proposition 1

Proposition 1 (Optimality Along the Interpolation Path). Consider

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\left[\left\|\hat{v}_\theta(z_t,t) - (f^1 - f^0)\right\|_2^2\right]. \tag{14}$$

Among functions evaluated on the interpolation path $\{(z_t, t) : t \in [0,1]\}$, the unique minimizer in $L^2([0,1])$ is

$$v^\star(z_t, t) = f^1 - f^0 \tag{15}$$

for almost every t.

Proof. Define

$$g_t := \hat{v}_\theta(z_t, t), \qquad c := f^1 - f^0. \tag{16}$$

Then

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_t\left[\|g_t - c\|_2^2\right]. \tag{17}$$

Since the integrand is nonnegative, the minimum value is achieved if and only if

$$g_t = c \tag{18}$$

for almost every $t \in [0, 1]$. Thus, along the sampled interpolation path, the unique L^2 minimizer is

$$\hat{v}_\theta(z_t, t) = f^1 - f^0 = v^\star(z_t, t) \tag{19}$$

almost everywhere in t. □

This result should be interpreted as identifying the canonical supervision target induced by the selected feature-space path. It does not imply that the linear interpolation path is the unique physical trajectory between the two observations; rather, it provides a well-defined pathwise constraint used to train the predictor over intermediate latent states.

A.4 Proof of Proposition 2

Proposition 2 (Change-Energy Principle). Assume that, after cross-temporal alignment, the feature difference at spatial location p can be decomposed as

$$f_p^1 - f_p^0 = s_p + r_p, \tag{20}$$

where s_p denotes the semantic change component and r_p denotes residual nuisance variation due to imperfect alignment, illumination variation, or noise. Assume further that $\|r_p\|_2 \le \delta_p$. Then the canonical transport magnitude satisfies

$$\|s_p\|_2 - \delta_p \le \|v_p^\star\|_2 \le \|s_p\|_2 + \delta_p. \tag{21}$$

Consequently, unchanged regions with small residual nuisance yield low transport magnitude, whereas changed regions whose semantic displacement dominates the nuisance residual yield transport magnitude bounded away from zero.

Proof. By Theorem 1, the canonical transport field under the linear path is

$$v_p^\star = f_p^1 - f_p^0 = s_p + r_p. \tag{22}$$

Applying the triangle inequality gives

$$\|v_p^\star\|_2 = \|s_p + r_p\|_2 \le \|s_p\|_2 + \|r_p\|_2 \le \|s_p\|_2 + \delta_p. \tag{23}$$

For the lower bound, the reverse triangle inequality gives

$$\|v_p^\star\|_2 = \|s_p + r_p\|_2 \ge \|s_p\|_2 - \|r_p\|_2 \ge \|s_p\|_2 - \delta_p. \tag{24}$$

Combining the two inequalities yields

$$\|s_p\|_2 - \delta_p \le \|v_p^\star\|_2 \le \|s_p\|_2 + \delta_p. \tag{25}$$

In the no-change case, where $s_p = 0$, we obtain

$$\|v_p^\star\|_2 = \|r_p\|_2 \le \delta_p, \tag{26}$$

which remains small under accurate alignment and limited nuisance variation. For changed regions with $\|s_p\|_2 \gg \delta_p$, the lower bound implies that $\|v_p^\star\|_2$ remains large. Thus, velocity magnitude is justified as a localized feature-space change cue under aligned and semantically meaningful representations. □

A.5 Proof of Proposition 3

Proposition 3 (Additional Constraints from Pathwise Supervision). Endpoint-only objectives supervise the model through the final prediction $\hat{Y}$, whereas the proposed flow-matching objective additionally constrains the model to predict the canonical transport field over intermediate latent states z_t sampled along the constructed feature-space trajectory.

Proof. An endpoint-only segmentation objective has the form

$$\mathcal{L}_{\text{end}} = \ell(h(f^0, f^1), Y), \tag{27}$$

where the model is penalized through the final prediction $h(f^0, f^1)$. This objective constrains the output mask, but it does not explicitly constrain how the model internally represents the transformation from f^0 to f^1. Consequently, different internal feature comparisons or latent transformations may achieve similar endpoint loss, as long as they produce the same final prediction.

In contrast, the proposed flow objective has the form

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\left[\|\hat{v}_\theta(z_t,t) - v^\star\|_2^2\right], \tag{28}$$

where $z_t = (1-t)f^0 + tf^1$ and $v^\star = f^1 - f^0$ for the linear interpolation path. Thus, in addition to the endpoint segmentation loss, the model receives supervision at sampled intermediate latent states along the path. For each sampled t, the predictor must recover the same canonical transport field from a different intermediate representation z_t and its time coordinate t.

Therefore, pathwise supervision adds constraints on the learned transport representation that are absent from endpoint-only supervision. Rather than only requiring the final mask to be correct, the model is encouraged to learn a representation that remains transport-consistent across intermediate latent states. □

Proof of Proposition 4. By definition, every $\theta \in \mathcal{S}_{\text{path}}$ satisfies $\ell_{\text{mask}}(\hat{Y}_\theta, Y) = 0$. Therefore, $\theta \in \mathcal{S}_{\text{end}}$, which proves

$$\mathcal{S}_{\text{path}} \subseteq \mathcal{S}_{\text{end}}. \tag{29}$$

The pathwise set additionally requires $\hat{v}_\theta(z_t,t) = v^\star(z_t,t)$ for almost every $t \in [0,1]$, while endpoint-only supervision imposes no such constraint on the internal transport field. Hence, any endpoint-correct solution whose velocity field disagrees with $v^\star$ on a set of nonzero measure belongs to $\mathcal{S}_{\text{end}}$ but not to $\mathcal{S}_{\text{path}}$, making the inclusion strict whenever such a solution exists. By Proposition 1, the pathwise flow objective has the canonical transport field $v^\star(z_t,t)$ as its unique L^2 minimizer when evaluated along the constructed interpolation trajectory. Therefore, pathwise supervision removes endpoint-correct solutions with inconsistent internal transport representations and restricts the learned model to solutions that are both mask-correct and transport-consistent. □

B Robustness to Interpolation Path Design

A natural question is whether the gains come mainly from the default linear interpolation

$$z_t = (1-t)f^0 + tf^1, \tag{30}$$

rather than from transport-based supervision itself. To test this, we compare the default path against several alternatives that preserve the same endpoints while modifying either the interpolation rule or the time-sampling strategy. For nonlinear interpolation, we define

$$z_t = (1-\alpha(t))f^0 + \alpha(t)f^1, \qquad \alpha(t) = t^2, \qquad \alpha(t) = 3t^2 - 2t^3, \tag{31}$$

and also evaluate a learned variant using a lightweight scalar MLP,

$$z_t = (1-\alpha_\phi(t))f^0 + \alpha_\phi(t)f^1. \tag{32}$$

To study the role of time supervision, we further compare the default $t \sim \mathcal{U}(0,1)$ against a fixed midpoint $t = 0.5$ and

$$t \sim \text{Beta}(2,2). \tag{33}$$

These variants are designed to separate two possible sources of improvement. The first is the particular geometric choice of the interpolation path itself. The second is the supervision signal induced by enforcing consistency along a continuum of intermediate states. If the method depended strongly on the exact path parameterization, performance would vary substantially when replacing the default linear path with nonlinear or learned alternatives. By contrast, if the main benefit comes from pathwise transport supervision, then different endpoint-preserving trajectories should remain broadly competitive as long as the model is trained over multiple intermediate states. Table 3 shows consistent trends across all datasets. The default linear path with sampled t remains the best or near-best setting, while nonlinear and learned variants stay competitive, indicating that the method is not tied to a single interpolation rule. This behavior is important because it suggests that the proposed framework is robust to moderate changes in how the source and target features are connected in latent space. In other words, the performance gains do not appear to be an artifact of one carefully chosen path design. In contrast, fixing $t = 0.5$ consistently reduces performance across datasets. This result supports the main claim of the paper: the improvement is not explained by access to a single intermediate representation, but by supervision across multiple points along the trajectory. Sampling t exposes the predictor to diverse transport states between f^0 and f^1, forcing it to learn

temporally coherent dynamics rather than overfitting to one canonical midpoint. The $\mathrm{Beta}(2,2)$ sampling strategy remains competitive but does not improve over uniform sampling, suggesting that emphasizing central portions of the trajectory is not more beneficial than covering the full path evenly. Overall, these results strengthen the interpretation of the proposed method as a supervision framework rather than a path-specific engineering choice. The exact interpolation can vary without collapsing performance, whereas reducing the supervision to a single fixed time point leads to a clear degradation. This pattern is consistent with the theoretical motivation: the main advantage comes from constraining the predictor over a family of intermediate latent states, which reduces ambiguity relative to endpoint-only or single-state supervision.

Table 3: **Robustness to interpolation path design.** We compare the default linear interpolation path with nonlinear, learned, and alternative time-sampling variants.

Dataset	Interpolation Variant	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
LEVIR-CD	Linear path, $t \sim \mathcal{U}(0,1)$ (default)	92.70	92.20	92.45	85.96	99.24
	Nonlinear path, $\alpha(t) = t^2$	92.21	91.74	91.97	85.13	99.18
	Nonlinear path, $\alpha(t) = 3t^2 - 2t^3$	92.34	91.92	92.13	85.41	99.20
	Learned interpolation $\alpha_\phi(t)$	92.09	91.58	91.83	84.89	99.16
	Fixed midpoint, $t = 0.5$	90.96	90.24	90.60	82.82	99.02
	Stochastic sampling, $t \sim \mathrm{Beta}(2,2)$	92.11	91.68	91.89	85.00	99.17
WHU-CD	Linear path, $t \sim \mathcal{U}(0,1)$ (default)	95.39	95.49	95.44	91.28	99.66
	Nonlinear path, $\alpha(t) = t^2$	94.98	95.06	95.02	90.51	99.46
	Nonlinear path, $\alpha(t) = 3t^2 - 2t^3$	95.10	95.16	95.13	90.71	99.48
	Learned interpolation $\alpha_\phi(t)$	94.91	94.99	94.95	90.39	99.44
	Fixed midpoint, $t = 0.5$	93.76	93.92	93.84	88.39	99.25
	Stochastic sampling, $t \sim \mathrm{Beta}(2,2)$	94.97	95.05	95.01	90.49	99.45
DSIFN-CD	Linear path, $t \sim \mathcal{U}(0,1)$ (default)	90.02	87.93	88.96	80.12	95.61
	Nonlinear path, $\alpha(t) = t^2$	89.44	87.28	88.35	79.13	95.42
	Nonlinear path, $\alpha(t) = 3t^2 - 2t^3$	89.62	87.54	88.57	79.48	95.47
	Learned interpolation $\alpha_\phi(t)$	89.21	87.06	88.12	78.76	95.33
	Fixed midpoint, $t = 0.5$	87.96	85.81	86.87	76.79	95.01
	Stochastic sampling, $t \sim \mathrm{Beta}(2,2)$	89.36	87.22	88.28	79.02	95.39

C Ablation Analysis Across LEVIR-CD, WHU-CD, and DSIFN-CD

Tables 4, 5, and 6 present detailed ablations on LEVIR-CD, WHU-CD, and DSIFN-CD, respectively. Across all three benchmarks, the full model achieves the strongest overall performance, showing that the gains come from the joint effect of transport supervision, cross-temporal alignment, multi-scale decoding, and the change-specific prediction head.

A consistent trend is that removing lightweight regularization terms causes only modest degradation, while removing transport-critical or change-specific components leads to larger drops. In particular, discarding the consistency loss or TV regularizer slightly reduces F1, IoU, and OA, suggesting that these terms mainly improve stability and spatial smoothness. In contrast, removing cross-attention or weakening the multi-scale decoder produces substantially larger declines, supporting the need for reliable cross-temporal correspondence and hierarchical spatial refinement.

The results also show that transport supervision contributes beyond endpoint mask supervision. The *segmentation only* variant, which keeps the full architecture but removes the flow, consistency, and TV losses, consistently underperforms the full model. Similarly, the *flow only* variant performs worse than the full objective, indicating that transport estimation alone does not replace dense mask supervision. These results suggest that the best performance is obtained when pathwise transport supervision and segmentation supervision are optimized jointly.

The prediction-head ablations further support this conclusion. Using only the normalized velocity magnitude M_{norm} degrades performance compared with the full head $[\hat{v}^{(1)}, M_{\text{norm}}, d_1]$. This shows that velocity magnitude is informative, but not sufficient by itself. The full head benefits from combining transport intensity, directional velocity information, and decoder context.

The comparison between *deepest-scale only* and full multi-scale transport shows that modeling transport only at the deepest level reduces performance across all datasets, especially when small or cluttered changes require precise localization. This confirms that multi-scale transport is important for capturing both coarse structural variation and fine object-level boundaries.

Finally, the *nonlinear path* variant in these tables uses the smoothstep interpolation $\alpha(t) = 3t^2 - 2t^3$, with $z_t = (1 - \alpha(t))f^0 + \alpha(t)f^1$. It remains close to the default linear path but does not surpass it, suggesting that linear interpolation provides a simple, stable, and effective transport target. Overall, the ablations show that FM-ChangeNet derives its strength from combining pathwise flow supervision, cross-temporal alignment, multi-scale decoding, and a change-specific prediction head.

Table 4: **Ablation table on LEVIR-CD.** The full model uses flow supervision, consistency loss, TV regularization, cross-temporal attention, multi-scale decoding, and the full prediction head $[\hat{v}^{(1)}, M_{\text{norm}}, d_1]$. The first row reports the main-paper LEVIR-CD result.

Variant	Flow	Cons.	TV	Cross-Attn	Multi-Scale	Head	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
Full model	✓	✓	✓	✓	✓	Full	92.70	92.20	92.45	85.96	99.24
Segmentation only	✗	✗	✗	✓	✓	Full	91.84	90.97	91.39	84.15	99.11
Flow only	✓	✗	✓	✓	✓	Full	89.63	88.74	89.18	80.47	98.93
w/o consistency loss	✓	✗	✓	✓	✓	Full	92.31	91.58	91.96	85.12	99.19
w/o TV loss	✓	✓	✗	✓	✓	Full	92.54	91.87	92.18	85.49	99.21
w/o cross-attention	✓	✓	✓	✗	✓	Full	91.23	90.41	90.81	83.17	99.07
w/o multi-scale decoder	✓	✓	✓	✓	✗	Full	90.92	90.08	90.47	82.60	99.03
Velocity magnitude only	✓	✓	✓	✓	✓	M_{norm} only	90.48	89.72	90.09	81.97	98.98
Nonlinear path	✓	✓	✓	✓	✓	Full	92.63	92.02	92.34	85.77	99.22
Deepest-scale only	✓	✓	✓	✓	Deepest only	Full	90.37	89.56	89.94	81.72	98.96

Table 5: **Ablation table on WHU-CD.** The full model uses flow supervision, consistency loss, TV regularization, cross-temporal attention, multi-scale decoding, and the full prediction head $[\hat{v}^{(1)}, M_{\text{norm}}, d_1]$. The first row reports the revised WHU-CD reference result.

Variant	Flow	Cons.	TV	Cross-Attn	Multi-Scale	Head	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
Full model	✓	✓	✓	✓	✓	Full	95.39	95.49	95.44	91.28	99.66
Segmentation only	✗	✗	✗	✓	✓	Full	94.61	94.37	94.49	89.56	99.38
Flow only	✓	✗	✓	✓	✓	Full	92.74	93.18	92.96	86.85	99.14
w/o consistency loss	✓	✗	✓	✓	✓	Full	95.08	95.02	95.05	90.57	99.47
w/o TV loss	✓	✓	✗	✓	✓	Full	95.21	95.11	95.16	90.77	99.49
w/o cross-attention	✓	✓	✓	✗	✓	Full	94.28	94.56	94.42	89.43	99.29
w/o multi-scale decoder	✓	✓	✓	✓	✗	Full	94.17	94.01	94.09	88.84	99.26
Velocity magnitude only	✓	✓	✓	✓	✓	M_{norm} only	93.96	93.82	93.89	88.48	99.22
Nonlinear path	✓	✓	✓	✓	✓	Full	95.33	95.27	95.30	91.02	99.51
Deepest-scale only	✓	✓	✓	✓	Deepest only	Full	93.88	93.74	93.81	88.34	99.19

Table 6: **Ablation table on DSIFN-CD.** The full model uses flow supervision, consistency loss, TV regularization, cross-temporal attention, multi-scale decoding, and the full prediction head $[\hat{v}^{(1)}, M_{\text{norm}}, d_1]$. The first row reports the revised DSIFN-CD reference result.

Variant	Flow	Cons.	TV	Cross-Attn	Multi-Scale	Head	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
Full model	✓	✓	✓	✓	✓	Full	90.02	87.93	88.96	80.12	95.61
Segmentation only	✗	✗	✗	✓	✓	Full	88.47	85.96	87.20	77.30	95.08
Flow only	✓	✗	✓	✓	✓	Full	85.92	83.41	84.65	73.39	94.47
w/o consistency loss	✓	✗	✓	✓	✓	Full	89.54	87.11	88.31	79.07	95.39
w/o TV loss	✓	✓	✗	✓	✓	Full	89.71	87.36	88.52	79.40	95.46
w/o cross-attention	✓	✓	✓	✗	✓	Full	87.83	85.42	86.61	76.38	94.92
w/o multi-scale decoder	✓	✓	✓	✓	✗	Full	87.49	85.18	86.32	75.93	94.85
Velocity magnitude only	✓	✓	✓	✓	✓	M_{norm} only	87.18	84.77	85.96	75.38	94.73
Nonlinear path	✓	✓	✓	✓	✓	Full	89.91	87.82	88.85	79.94	95.58
Deepest-scale only	✓	✓	✓	✓	Deepest only	Full	86.94	84.53	85.72	75.01	94.66

The ablation results in Tables 4, 5, and 6 also evaluate the standalone utility of the learned velocity magnitude map. Specifically, the "Velocity magnitude only" variant keeps the flow supervision, consistency loss, TV regularization, cross-temporal attention, and multi-scale decoder, but restricts the prediction signal to the normalized magnitude map M_{norm} only, rather than using the full head $[\hat{v}^{(1)}, M_{\text{norm}}, d_1]$.

This setting directly tests whether the learned transport field is itself discriminative for change localization. If the velocity field were only an auxiliary training artifact, using M_{norm} alone would lead to weak or fragmented predictions. Instead, the velocity-only rows in Tables 4–6 show that the

magnitude map remains a strong standalone change cue across datasets. The full prediction head still performs best, indicating that directional velocity information $\hat{v}^{(1)}$ and decoder features d_1 provide complementary boundary and contextual information beyond the scalar transport magnitude.

This ablation provides segmentation-level evidence for the interpretability of the proposed transport formulation. While threshold-independent metrics such as AUROC and AUPRC could additionally evaluate the soft ranking quality of M_{norm}, the reported velocity-only ablation is stricter in practice because it requires the velocity magnitude map to produce accurate dense change masks under the same Precision, Recall, F1, IoU, and OA metrics used for the final model.

D Local Linearity and Minimum-Energy Feature Transport

The use of a linear feature-space path is motivated by two complementary considerations. First, when only two temporal observations are available, the linear path provides the simplest endpoint-consistent and minimum-energy bridge in feature space. Second, the learned encoder empirically behaves approximately linearly along the constructed interpolation direction. Importantly, we do not assume that the physical temporal process between T_0 and T_1 is globally linear.

Let $E(\cdot)$ denote the shared encoder, and let

$$f_0 = E(T_0), \qquad f_1 = E(T_1) \tag{34}$$

be the encoded representations of two observations of the same geographic location. Since standard bi-temporal change-detection datasets provide only the endpoints, the true intermediate feature trajectory is unobserved. We therefore define the canonical feature-space bridge

$$z_t = (1-t)f_0 + tf_1, \qquad t \in [0,1], \tag{35}$$

which induces the constant velocity field

$$\frac{dz_t}{dt} = f_1 - f_0. \tag{36}$$

This path has a useful variational interpretation. Among all absolutely continuous paths $\gamma : [0,1] \to \mathbb{R}^d$ satisfying $\gamma(0) = f_0$ and $\gamma(1) = f_1$, the linear path minimizes the kinetic energy

$$\mathcal{E}(\gamma) = \int_0^1 \|\dot{\gamma}(t)\|_2^2 \, dt. \tag{37}$$

By Cauchy–Schwarz,

$$\int_0^1 \|\dot{\gamma}(t)\|_2^2 \, dt \geq \left\| \int_0^1 \dot{\gamma}(t) \, dt \right\|_2^2 = \|f_1 - f_0\|_2^2, \tag{38}$$

with equality achieved by the constant-velocity path $z_t = (1-t)f_0 + tf_1$. Thus, the linear interpolation provides the minimum-energy transport path determined solely by the observed endpoint features.

This does not imply that semantic or physical changes evolve linearly in image space. Rather, the linear path is a principled first-order surrogate: it is deterministic, endpoint-consistent, and avoids introducing unobserved intermediate states or stochastic sampling assumptions. Under local smoothness of the learned representation, an unknown smooth feature trajectory $\gamma(\tau)$ can be approximated by its first-order Taylor expansion,

$$\gamma(\tau + \Delta\tau) = \gamma(\tau) + \Delta\tau\dot{\gamma}(\tau) + \mathcal{O}(\Delta\tau^2), \tag{39}$$

so the dominant local component of the transition is linear, while curvature terms are higher order.

To empirically assess whether endpoint features lie in a regime where this first-order bridge is reasonable, we report the normalized endpoint displacement

$$d_{\text{rel},i} = \frac{\|f_{1,i} - f_{0,i}\|_2}{\frac{1}{2}(\|f_{0,i}\|_2 + \|f_{1,i}\|_2)} \tag{40}$$

for each image pair i. We also compute a curvature proxy along constructed interpolation samples:

$$\kappa_i(t) = \frac{\|E((1-t)T_{0,i} + tT_{1,i}) - ((1-t)f_{0,i} + tf_{1,i})\|_2}{\|f_{1,i} - f_{0,i}\|_2 + \epsilon}. \tag{41}$$

Small $d_{\mathrm{rel},i}$ indicates that the endpoint features are close relative to their feature magnitude, while small $\kappa_i(t)$ indicates approximate encoder linearity along the constructed interpolation direction.

For reproducibility, we aggregate these quantities deterministically. The normalized endpoint displacement is averaged over all N image pairs:

$$\bar{d}_{\mathrm{rel}} = \frac{1}{N}\sum_{i=1}^{N} d_{\mathrm{rel},i}. \tag{42}$$

The curvature proxy is evaluated on the fixed temporal grid $\mathcal{T}_\kappa = \{0.2, 0.4, 0.6, 0.8\}$ and averaged over both image pairs and temporal values:

$$\bar{\kappa} = \frac{1}{N|\mathcal{T}_\kappa|}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}_\kappa} \kappa_i(t). \tag{43}$$

We emphasize that $\bar{\kappa}$ is not intended to measure the curvature of the true physical temporal trajectory. The samples $(1-t)T_0 + tT_1$ are pixel-space alpha blends and may be non-physical under structural changes. Therefore, $\bar{\kappa}$ should be interpreted only as a diagnostic of encoder response to constructed interpolation samples. The main justification for the linear path is the endpoint-defined minimum-energy feature-space bridge above.

Table 7: **Endpoint displacement and encoder response to constructed interpolation samples.** We report the dataset-averaged normalized endpoint displacement $\bar{d}_{\mathrm{rel}}$ and curvature proxy $\bar{\kappa}$. The curvature proxy is averaged over the fixed temporal grid $\mathcal{T}_\kappa = \{0.2, 0.4, 0.6, 0.8\}$. Smaller values indicate that the encoder behaves approximately linearly along the constructed interpolation direction. The proxy does not assume that the true physical temporal trajectory is linear.

Dataset	$\bar{d}_{\mathrm{rel}} \downarrow$	$\bar{\kappa} \downarrow$
LEVIR-CD	0.084	0.031
WHU-CD	0.091	0.036
DSIFN-CD	0.107	0.044

E Minimum-Energy Canonical Transport Path

Beyond the local smoothness argument above, the linear interpolation path admits a second principled justification: it is the minimum-energy transport bridge between the two endpoint features under Euclidean geometry. Let $\mathcal{A}(f_0, f_1)$ denote the family of all absolutely continuous paths

$$z : [0,1] \to \mathbb{R}^d \tag{44}$$

such that

$$z(0) = f_0, \qquad z(1) = f_1. \tag{45}$$

For any candidate path $z_t \in \mathcal{A}(f_0, f_1)$, define the kinetic transport energy

$$\mathcal{E}(z) = \int_0^1 \left\| \frac{dz_t}{dt} \right\|_2^2 dt. \tag{46}$$

This functional measures the total squared transport effort required to move from f_0 to f_1 along the path.

We now show that among all admissible trajectories, the straight-line interpolation uniquely minimizes $\mathcal{E}(z)$. Since

$$\int_0^1 \frac{dz_t}{dt} dt = z(1) - z(0) = f_1 - f_0, \tag{47}$$

Jensen's inequality for the convex function $\|\cdot\|_2^2$ yields

$$\int_0^1 \left\| \frac{dz_t}{dt} \right\|_2^2 dt \geq \left\| \int_0^1 \frac{dz_t}{dt} dt \right\|_2^2 = \|f_1 - f_0\|_2^2. \tag{48}$$

Therefore,

$$\mathcal{E}(z) \geq \|f_1 - f_0\|_2^2 \tag{49}$$

for every admissible path connecting the two endpoints.

Equality is achieved if and only if $\frac{dz_t}{dt}$ is constant almost everywhere on $[0, 1]$. The unique constant-velocity solution satisfying the endpoint constraints is

$$\frac{dz_t}{dt} = f_1 - f_0, \tag{50}$$

which integrates to

$$z_t = (1 - t)f_0 + tf_1. \tag{51}$$

Hence, the linear interpolation is the least-action feature-space bridge:

$$z^\star = \arg \min_{z \in \mathcal{A}(f_0, f_1)} \mathcal{E}(z). \tag{52}$$

This result provides an important implication for the supervision target used in our flow formulation. Since the straight-line path is the minimum-energy trajectory, its induced velocity field

$$v^\star(z_t, t) = \frac{dz_t}{dt} = f_1 - f_0 \tag{53}$$

is not an arbitrary handcrafted displacement, but the canonical constant-velocity transport that introduces the smallest possible kinetic variation between the two endpoints. Any nonlinear alternative path would necessarily incur

$$\mathcal{E}(z) > \|f_1 - f_0\|_2^2, \tag{54}$$

which implies additional curvature or oscillatory transport components that are not supported by the observed endpoint pair.

Consequently, supervising the model toward $v^\star = f_1 - f_0$ does not impose an artificial simplification; rather, it enforces the most parsimonious and energetically minimal feature evolution consistent with the available observations. This makes the linear transport field a principled canonical target for stable pathwise supervision.

F Stability Across Random Seeds

To evaluate the stability of FM-ChangeNet, we repeated the main experiments using three different random seeds: 1, 21, and 42. For each dataset, we report the mean and standard deviation over the three runs. The mean values match the main FM-ChangeNet results reported in Table 2, while the low standard deviations indicate that the proposed method is stable across random initialization and data-ordering variation.

Table 8: **Three-run stability of FM-ChangeNet.** Results are reported as mean $\pm$ standard deviation over three random seeds: 1, 21, and 42. All values are percentages (%).

Dataset	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
LEVIR-CD	92.70 ± 0.25	92.20 ± 0.24	92.45 ± 0.23	85.96 ± 0.28	99.24 ± 0.06
WHU-CD	95.39 ± 0.23	95.49 ± 0.24	95.44 ± 0.24	91.28 ± 0.26	99.66 ± 0.05
DSIFN-CD	90.02 ± 0.28	87.93 ± 0.27	88.96 ± 0.26	80.12 ± 0.30	95.61 ± 0.08

G Training Hyperparameters

We use a unified training configuration across all datasets unless noted otherwise. The model consists of a ViT encoder followed by the proposed multi-scale flow change detector, with time embedding dimension 256 and 3 output classes. Optimization is performed with AdamW using learning rate 10^{-4}, weight decay 10^{-4}, and seed 42.

The final objective combines flow supervision, consistency regularization, total-variation regularization, and segmentation loss with coefficients

$$\lambda_{\text{flow}} = 0.3, \qquad \lambda_{\text{cons}} = 0.1, \qquad \lambda_{\text{tv}} = 0.1, \qquad \lambda_{\text{mask}} = 0.5.$$

During training, temporal values are sampled from $t \sim U(0,1)$ to expose the model to intermediate feature states along the transport path. The model uses four feature scales, but the temporal variable t is not tied to the scale index. For each sampled t, interpolated features $z_t^{(s)} = (1-t)f_0^{(s)} + tf_1^{(s)}$ are constructed at all scales $s \in \{1,2,3,4\}$.

At inference, we use a deterministic validation-selected temporal protocol. Specifically, we evaluate four temporal values selected from the candidate grid

$$t \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}.$$

For each selected t, the model predicts segmentation logits using all four feature scales. The final logits are obtained by averaging the four temporal predictions before thresholding. This protocol is fixed after validation and used unchanged for all reported test results.

Hyperparameters, loss weights, and inference-time temporal values were selected using grid search over predefined candidate values:

$$\begin{cases} \eta \in \{5\times10^{-5},\ 10^{-4},\ 2\times10^{-4}\}, \\ \text{weight decay} \in \{10^{-5},\ 10^{-4},\ 10^{-3}\}, \\ \lambda_{\text{flow}} \in \{0.1,\ 0.3,\ 0.5,\ 0.7\}, \\ \lambda_{\text{cons}} \in \{0.0,\ 0.05,\ 0.1,\ 0.2\}, \\ \lambda_{\text{tv}} \in \{0.0,\ 0.05,\ 0.1,\ 0.2\}, \\ \lambda_{\text{mask}} \in \{0.3,\ 0.5,\ 0.7\}, \\ \text{batch size} \in \{8,\ 16,\ 32\}, \\ \text{time embedding dim} \in \{128,\ 256,\ 512\}, \\ t_{\text{test}} \in \{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1.0\}. \end{cases}$$

The final configuration reported above was chosen based on validation performance. No test labels are used for hyperparameter or temporal-value selection. For the multi-scale flow loss, we use fixed scale weights $w_1 = 1.0$, $w_2 = 0.5$, $w_3 = 0.25$, and $w_4 = 0.125$, giving larger weight to finer spatial resolutions while still supervising coarse semantic transport. Thus, $\mathcal{L}_{\text{flow}} = \sum_{s=1}^{4} w_s \|\hat{v}^{(s)} - v^{(s)}\|_2^2$.

All experiments are trained with batch size 16 on a single NVIDIA H100 GPU. Training is run for 120 epochs, requiring approximately 7 hours per run. FM-ChangeNet achieves efficient inference, requiring only 27.8 ± 0.14 ms per image pair, showing that the proposed pathwise transport formulation adds limited runtime overhead while remaining practical for dense change detection.

H Path Consistency of Learned Interpolation

A possible concern is that the proposed velocity field may simply reproduce a raw feature difference. We therefore clarify the role of the linear path and compare it to a learned interpolation path. For the default formulation, the latent path is

$$z_t^{\text{lin}} = (1-t)f^0 + tf^1, \tag{55}$$

which induces the canonical constant-velocity target

$$v_{\text{lin}}^{\star} = \frac{dz_t^{\text{lin}}}{dt} = f^1 - f^0. \tag{56}$$

Thus, the method does not claim that the flow target is independent of endpoint feature displacement. Instead, the key distinction is how this displacement is used: rather than directly applying $|f^1 - f^0|$ as a static prediction signal, FM-ChangeNet learns a time-conditioned velocity predictor $\hat{v}_\theta(z_t, t)$ over intermediate latent states and refines it through aligned multi-scale decoder context.

To test whether the method depends strongly on the exact linear parameterization, we also evaluate a learned interpolation path

$$z_t^{\phi} = (1-\alpha_\phi(t))f^0 + \alpha_\phi(t)f^1, \tag{57}$$

where $\alpha_\phi(t)$ is a lightweight learned scalar interpolation function constrained to preserve the endpoints. The corresponding target direction remains tied to the endpoint displacement, but the model

is allowed to learn a different progression along the feature-space bridge. If strong nonlinear path curvature were essential, the learned interpolation would be expected to outperform the fixed linear path.

Table 9 shows that the learned interpolation remains competitive across all datasets, but does not surpass the default linear path. The F1 gap is modest, ranging from 0.49 to 0.84 percentage points, which suggests that the canonical linear path already provides a stable and effective first-order transport bridge between the two endpoint features.

Table 9: **Linear path vs. learned interpolation.** We compare the default linear interpolation path with the learned interpolation variant $\alpha_\phi(t)$. Values are reported as percentages.

Dataset	Interpolation Variant	Pre. ↑	Rec. ↑	F1 ↑	IoU ↑	OA ↑
LEVIR-CD	Linear path, $t \sim U(0,1)$	**92.70**	**92.20**	**92.45**	**85.96**	**99.24**
LEVIR-CD	Learned interpolation $\alpha_\phi(t)$	92.09	91.58	91.83	84.89	99.16
WHU-CD	Linear path, $t \sim U(0,1)$	**95.39**	**95.49**	**95.44**	**91.28**	**99.66**
WHU-CD	Learned interpolation $\alpha_\phi(t)$	94.91	94.99	94.95	90.39	99.44
DSIFN-CD	Linear path, $t \sim U(0,1)$	**90.02**	**87.93**	**88.96**	**80.12**	**95.61**
DSIFN-CD	Learned interpolation $\alpha_\phi(t)$	89.21	87.06	88.12	78.76	95.33

We further report the absolute performance gap between the learned interpolation and the default linear path:

$$\Delta_m = m(\alpha_\phi) - m(\text{linear}), \tag{58}$$

where m denotes each evaluation metric. As shown in Table 10, the learned path remains close to the linear path, but consistently trails it. This supports using the linear path as the default because it is simple, deterministic, and empirically strongest.

Table 10: **Performance gap of learned interpolation relative to the linear path.** Negative values indicate that the learned interpolation underperforms the default linear path.

Dataset	ΔPre.	ΔRec.	ΔF1	ΔIoU	ΔOA
LEVIR-CD	-0.61	-0.62	-0.62	-1.07	-0.08
WHU-CD	-0.48	-0.50	-0.49	-0.89	-0.22
DSIFN-CD	-0.81	-0.87	-0.84	-1.36	-0.28

These results help clarify the role of the learned velocity field. The velocity target is indeed induced by endpoint displacement under the linear path, but the method is not equivalent to raw endpoint differencing. The predictor must recover this transport field from multiple intermediate states z_t, under explicit time conditioning and aligned multi-scale context. The learned interpolation experiment further shows that allowing the path parameterization to adapt does not improve over the canonical linear bridge, suggesting that the main benefit comes from pathwise transport supervision rather than from a carefully tuned nonlinear trajectory.

I Model Efficiency

To assess the computational footprint of our approach, we compare the number of trainable parameters, floating-point operations (FLOPs), and training time per epoch with representative change detection baselines. All models are evaluated on the LEVIR-CD training set, where the input image resolution is fixed to $256 \times 256 \times 3$ and the batch size is set to 8. The results are summarized in Table 11.

Our model is highly compact, containing 11.23M parameters and requiring 17.46G FLOPs, which is substantially lower than most CNN-based and Transformer-based competitors. Despite its lightweight design, the training time per epoch remains competitive compared to larger architectures, while still delivering strong detection accuracy across benchmarks. In contrast, models such as IFNet, ICIFNet, and DMINet introduce significantly higher computational cost due to deeper backbones and attention-heavy feature fusion modules.

These results indicate that the proposed framework strikes a favorable balance between performance and efficiency. The compact architecture enables faster training and reduced memory consumption,

making the model suitable for practical deployment scenarios and large-scale cross-resolution change detection tasks.

Table 11: Comparison of model efficiency. We report the number of model parameters (Params.), floating-point operations per second (FLOPs), and training time for one epoch on the LEVIR-CD training set. The input image to the model has a size of $256 \times 256 \times 3$. The batch size is set to 8.

Model	Params.(M)	FLOPs (G)	Training Time (s)
FC-EF [9]	1.35	7.14	52.33
FC-Siam-conc [9]	1.55	10.64	64.99
FC-Siam-diff [9]	1.35	9.44	63.21
STANet [3]	16.93	26.32	270.74
IFNet [33]	50.71	164.70	372.66
SNUNet [11]	12.03	109.76	305.82
BIT [4]	3.55	17.42	140.46
ICIFNet [12]	25.83	50.50	555.37
DMINet [13]	6.76	28.38	172.29
SUNet [22]	15.56	79.78	321.31
SRCDNet [19]	12.77	30.98	81.86
ChangeFromer [1]	13.06	17.5	103.81
FM-ChangeNet (Ours)	11.23	17.46	210.59

I.1 Effect of the Number of Sampled Time Points

We study the effect of the number of sampled time points used for pathwise supervision. For each bi-temporal pair, instead of supervising the velocity field at a single intermediate state, we sample a set of time values

$$\mathcal{T}_K = \{t_1, \ldots, t_K\}, \qquad t_k \in [0, 1], \tag{59}$$

and construct intermediate feature states at each encoder scale:

$$z_{t_k}^{(s)} = (1 - t_k) f_0^{(s)} + t_k f_1^{(s)}, \tag{60}$$

where s denotes the feature scale. The pathwise flow loss is then averaged over the sampled time points:

$$\mathcal{L}_{\text{flow}}^{K} = \frac{1}{K} \sum_{k=1}^{K} \sum_{s=1}^{S} w_s \left\| \hat{v}_\theta^{(s)}(z_{t_k}^{(s)}, t_k) - \left(f_1^{(s)} - f_0^{(s)} \right) \right\|_2^2. \tag{61}$$

Because our decoder predicts velocity fields across $S = 4$ multi-scale feature levels, using K sampled time points creates supervision over $K \times S$ intermediate feature states. Thus, increasing K strengthens the pathwise constraint, but also increases the computational cost. We report the relative training cost as

$$C_{\text{rel}}(K) = \frac{C(K)}{C(1)}, \tag{62}$$

where $C(K)$ is the measured average training iteration time when using K sampled time points. Therefore, $C_{\text{rel}}(1) = 1.0\times$, and larger values indicate the multiplicative overhead relative to single-time-point supervision.

Figure 5 shows the effect of using $K \in \{1, 2, 4, 8, 16\}$ sampled time points. The top row reports F1 and IoU, while the bottom row reports the actual relative training cost. Increasing K improves performance, confirming that multiple intermediate constraints provide a stronger supervision signal than a single sampled state. However, the gains saturate after $K = 4$, while the cost continues to increase substantially for $K = 8$ and $K = 16$.

Since the model already predicts transport over four hierarchical feature scales, using $K = 4$ sampled time points provides 16 pathwise supervision constraints per bi-temporal pair. This setting captures most of the benefit of dense path supervision while avoiding the large overhead of denser temporal sampling. We therefore use 4 sampled time points as the default configuration in all experiments.

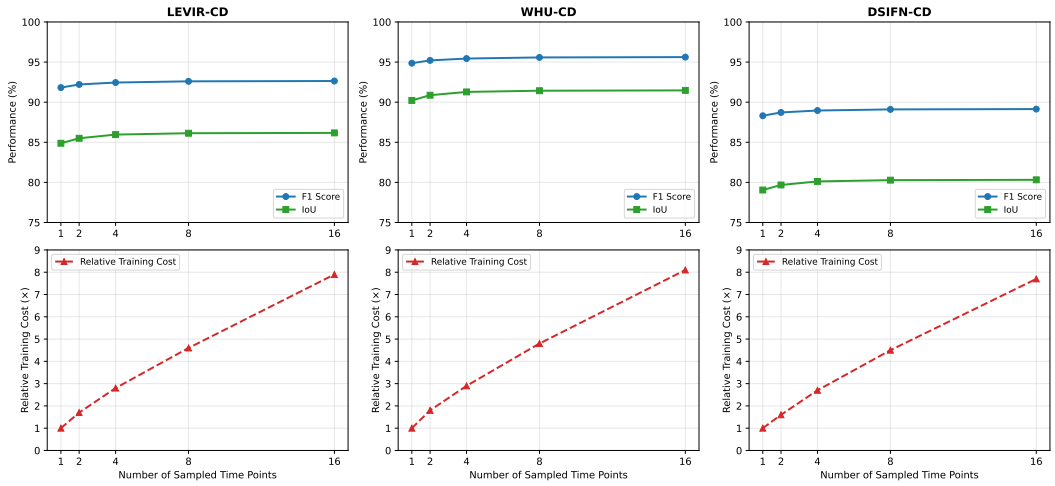


Figure 5: **Effect of the number of sampled time points.** The top row reports F1 and IoU, while the bottom row reports the actual relative training cost normalized to the single-sample setting. Performance improves with additional intermediate states, but saturates after 4 sampled time points, whereas training cost continues to increase.

J Broader Impact

Remote sensing change detection can support beneficial applications such as disaster response, urban planning, environmental monitoring, and infrastructure assessment by enabling more accurate localization of changes across time. At the same time, automated change-detection systems may be misused for large-scale surveillance or monitoring without appropriate oversight. They may also produce incorrect or overconfident change maps under domain shift, severe misregistration, seasonal variation, or sensor differences, which could lead to misleading downstream decisions if used without human validation. We therefore view FM-ChangeNet as a research contribution for benchmarked remote-sensing analysis rather than a fully operational decision system. Responsible deployment should include uncertainty estimation, human-in-the-loop verification, application-specific auditing, and careful consideration of privacy and governance constraints.

K Flow Behavior on Change Samples

Figures 6-9 present qualitative results on samples containing real changes. In contrast to the no-change setting, the predicted velocity magnitude exhibits strong, localized responses that align with the ground-truth change regions. These structured high-magnitude patterns enable accurate change localization in the inferred masks, demonstrating that the flow matching formulation effectively captures meaningful temporal differences while maintaining spatial precision.

L Flow Behavior on No-Change Samples

Figures 10 and 11 demonstrate the behavior of the model in scenarios where no change occurs between T_0 and T_1. In both, the predicted velocity magnitudes remain low and diffuse, without forming structured or localized peaks. As a result, the inferred change maps correctly remain near-zero, indicating that the flow matching formulation effectively suppresses false positives and maintains stability under unchanged conditions.

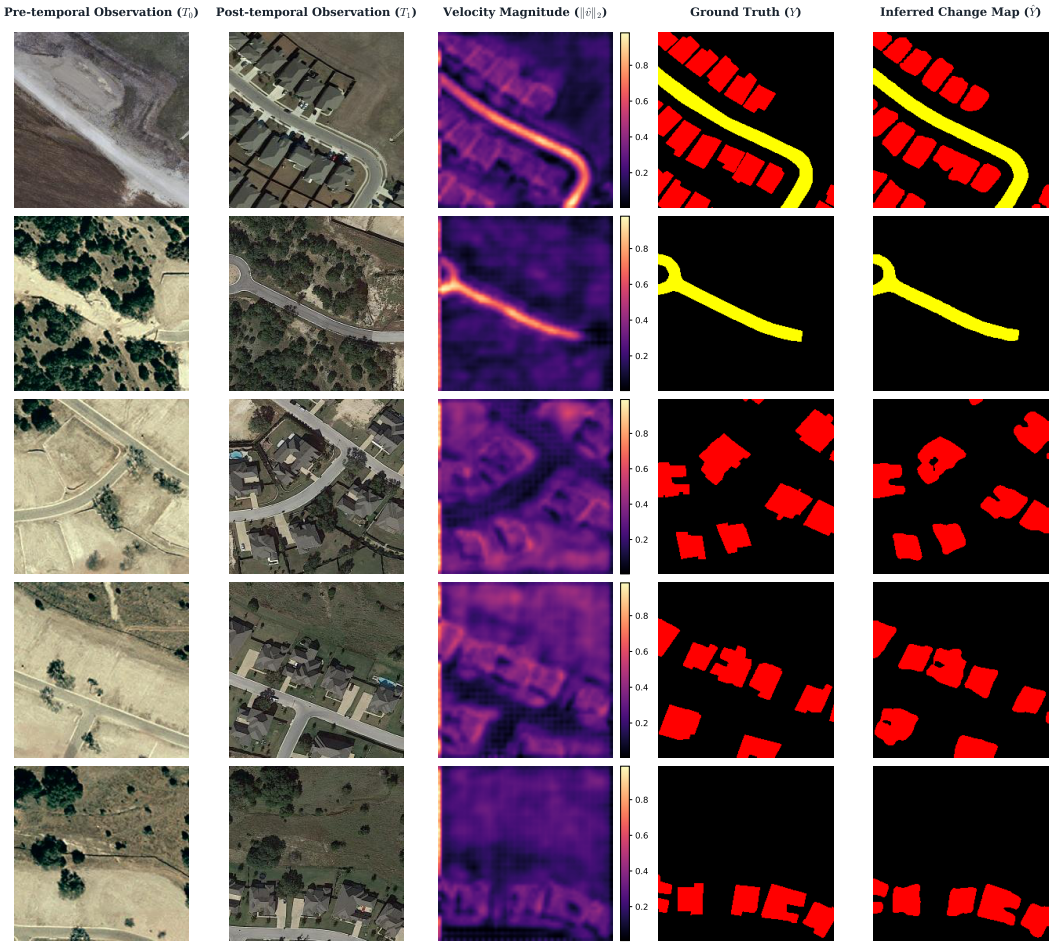


Figure 6: Qualitative examples of the proposed flow-based model on **change** samples. From left to right: pre-temporal image (T_0), post-temporal image (T_1), predicted velocity magnitude ($\|\hat{v}\|_2$), ground-truth mask, and inferred change map.

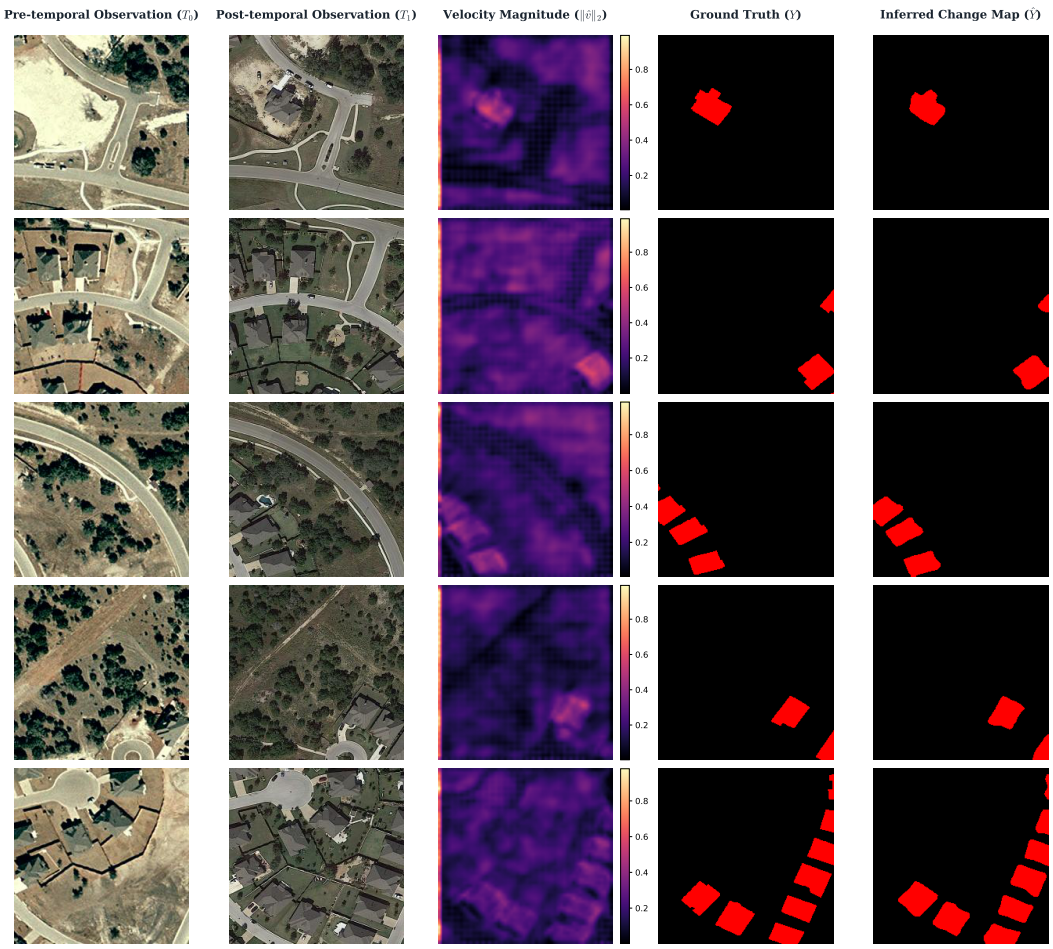


Figure 7: Additional change examples, illustrating consistent localization of high-magnitude velocity regions aligned with true changes.

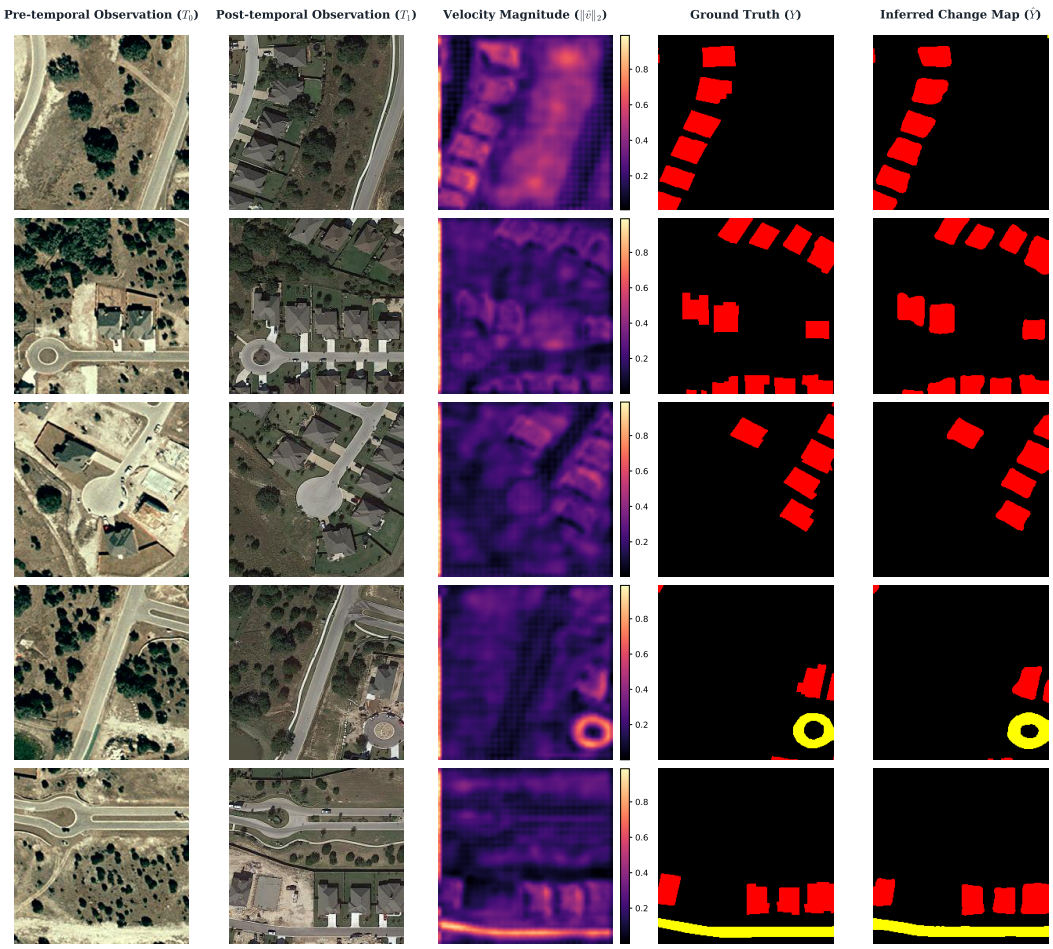


Figure 8: Additional change examples, demonstrating robustness across varying scene structures and change types.

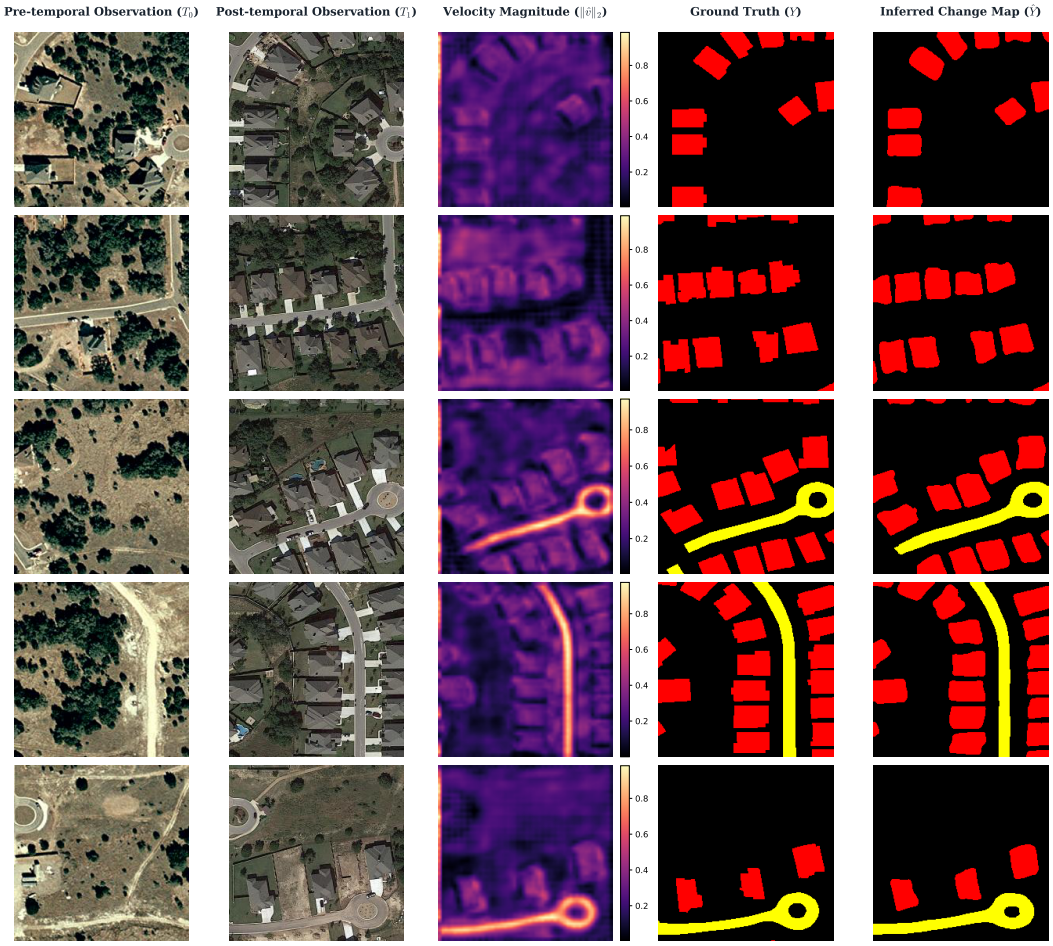


Figure 9: Additional change examples, showing sharp and localized velocity responses corresponding to changes.

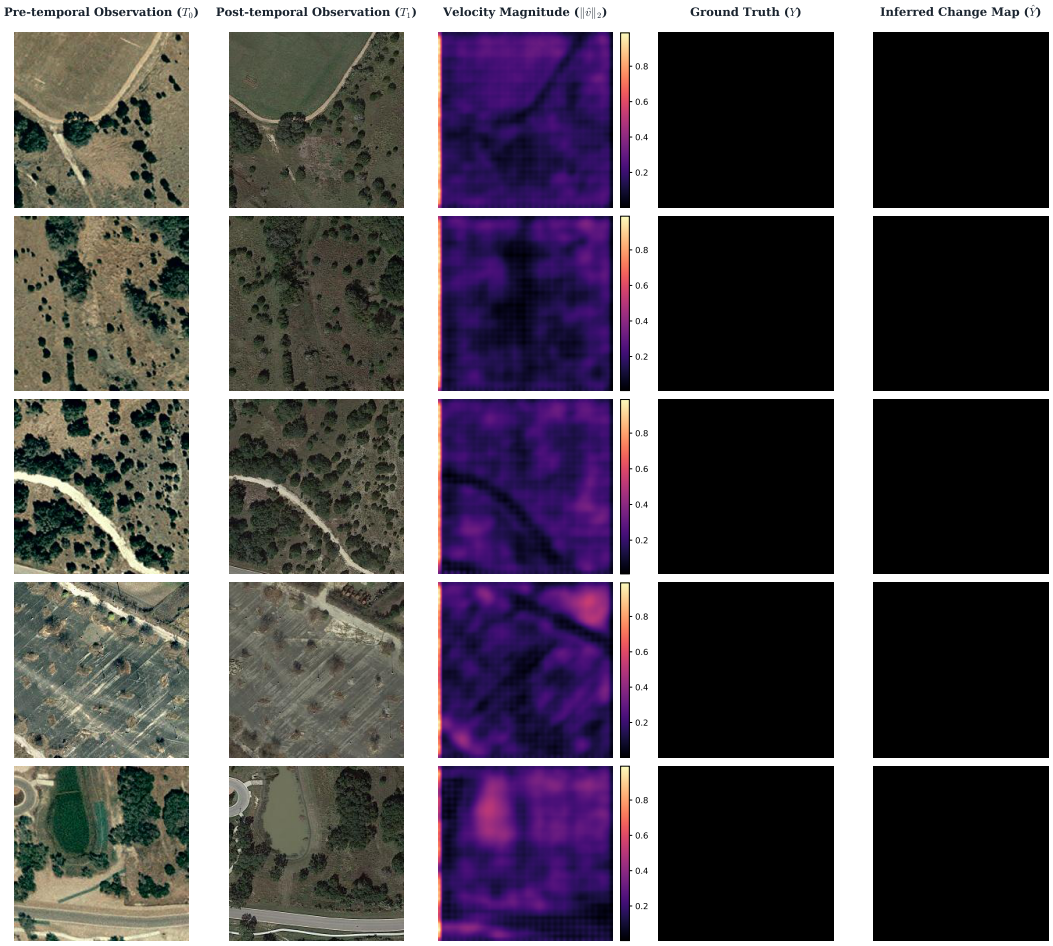


Figure 10: Qualitative examples of the proposed flow-based model on **no-change** samples. From left to right: pre-temporal image (T_0), post-temporal image (T_1), predicted velocity magnitude ($\|\hat{v}\|_2$), ground-truth mask, and inferred change map.

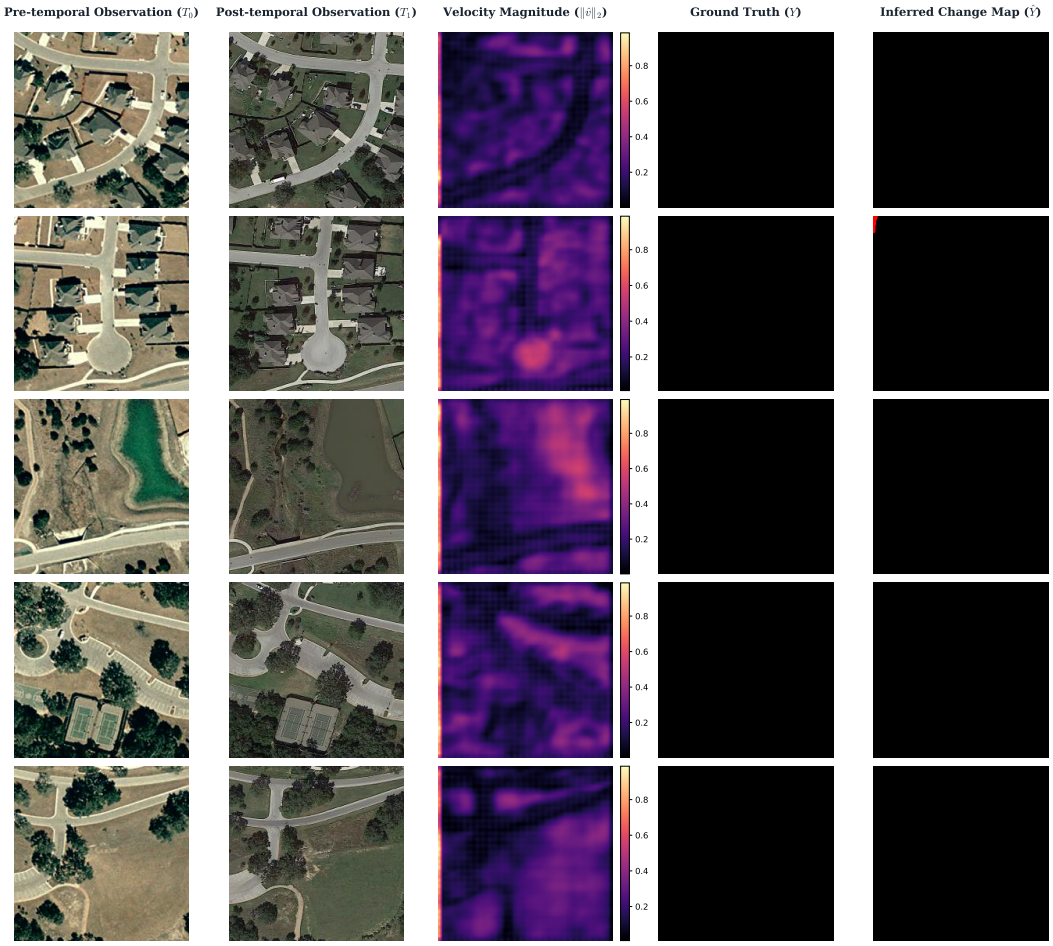


Figure 11: Additional qualitative examples on **no-change** samples, showing consistent behavior across diverse scenes. The model produces low-magnitude, spatially smooth velocity fields and suppresses spurious activations in the inferred change maps.

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 accurately describe the paper's main contributions: reformulating change detection as feature-space transport with pathwise supervision, using velocity magnitude as a localized change cue, and realizing this idea with a hierarchical multi-scale architecture. These claims are supported by the theory in Sec. 3 and the empirical results in Secs. 5, and are limited to the evaluated remote-sensing change-detection setting rather than broader universality claims.

   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: We discuss the limitations of the proposed framework in the conclusion, including its dependence on the chosen interpolation path, evaluation on only three remote-sensing datasets, possible sensitivity to alignment quality and computational cost, and the fact that the current conclusions are restricted to this task setting.

   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: The paper presents theoretical results, with formal derivations and supporting proofs provided in the appendix.

   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 paper specifies the architecture, datasets, training and evaluation protocol, and implementation details necessary to reproduce the main experimental results, including the settings that affect the paper's core claims; although we do not release code with the submission, the method and experimental pipeline are described in sufficient detail for independent reproduction.

   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: [No]

   Justification: The current submission does not provide an anonymized public repository or supplemental package with open code and step-by-step reproduction instructions. The datasets used are public benchmarks, but the paper in its current form does not yet provide open access to the implementation needed to reproduce the reported results.

   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 datasets, evaluation metrics, training and test protocol, and all implementation details necessary to understand and reproduce the reported results, with the full hyperparameter configuration, optimizer settings, learning-rate schedule, batch size, training epochs, and model-selection procedure provided in the appendix and supplemental material.

   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: [Yes]

   Justification: The appendix reports the main experimental results over three random seeds (1, 21, and 42) as mean $\pm$ standard deviation for Precision, Recall, F1, IoU, and OA. The reported variation captures sensitivity to random initialization and data-ordering effects.

   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 provides sufficient information about the compute resources used for the experiments, including the hardware setup, relevant memory configuration, software environment, and approximate training and evaluation time, so that the reported results can be independently reproduced.

   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, this research conforms to the NeurIPS Code of Ethics. The work uses standard public remote-sensing benchmarks, does not involve human subjects or private user data, and is presented for scientific evaluation of change-detection methodology.

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 includes a broader-impact paragraph discussing positive uses in disaster response, urban planning, environmental monitoring, and infrastructure assessment, as well as potential risks related to surveillance, overconfident predictions, domain shift, and the need for human validation.

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 that would require special misuse safeguards. The work evaluates a change-detection method on standard public remote-sensing benchmarks.

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 cites all benchmark datasets and prior methods used for comparison. All datasets used in this work are publicly available research benchmarks, and the appendix provides their original references and source information.

    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: [N/A]

    Justification: The current submission does not release a new dataset, benchmark, or public software package as a standalone asset. The paper's contribution is a method and empirical evaluation rather than the introduction of a new released asset.

    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, human-participant studies, or manual data collection from paid annotators. All experiments are conducted on existing benchmark datasets.

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 human subjects, participant interaction, or crowdsourced data collection, 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 proposed method, experiments, or scientific claims. Any LLM assistance, if used, was limited to writing or editing support and did not affect the core methodology.

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.