FLUIDRAIN: DIVERGENCE-FREE RAIN FLOW AS AN ATTENTION BIAS FOR LOOP-IN-LOOP VIDEO DERAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing video deraining methods typically exploit neighbouring frames through either explicit alignment or implicit spatiotemporal aggregation. Explicit alignment relies on accurate motion estimation, which can become unreliable under dense rain, while implicit aggregation avoids alignment but lacks explicit guidance on the directional and temporally coherent structure of rain. This leaves a gap between reliable temporal aggregation and explicit modeling of rain motion. We propose FluidRain, a parameter-efficient video derainer that uses divergence-free rain flow to guide **Loop-in-Loop** attention across scales and neighbouring frames. Motivated by fluid mechanics, FluidRain models rain motion as an image-space flow, projects the estimated field onto the divergence-free subspace, and converts its local direction and extent into an anisotropic attention bias. The same rain-aware operator is then reused across neighbouring frames and three resolutions in a Loop-in-Loop design. The resulting three-frame model has only **0.80M** parameters and remains competitive with substantially larger restoration networks on four benchmarks. Controlled view experiments show that the shared operator benefits most from complementary neighbouring observations rather than repeated computation. To evaluate whether the model remains reliable when rain motion changes across frames, we introduce RainSyn-Gust, which injects controlled changes in rain-streak direction into existing benchmarks. We also use a physics-inspired reference-free diagnostic to analyze removal on real rainy videos.

1 INTRODUCTION

Rain streaks in video arise from the motion blur of falling raindrops carried by the wind and imaged by a moving camera. Their apparent direction and length are jointly determined by wind, scene depth, and camera motion, and thus vary across sequences (Garg & Nayar, 2004).

Early video deraining methods were largely physics-driven. They modeled the formation and appearance of rain streaks (Garg & Nayar, 2004; 2006) and combined these models with hand-crafted temporal or low-rank priors (Kim et al., 2015). With the rise of deep learning, the field gradually shifted toward data-driven temporal modeling, first through recurrent architectures (Liu et al., 2018; Yang et al., 2021; Yue et al., 2021) and more recently through attention and state-space models (Wu et al., 2024; Sun et al., 2025). Modern methods typically exploit neighbouring frames in two ways. One line explicitly aligns them using optical flow or related motion estimation, whose reliability can degrade in heavily rain-corrupted regions (Chan et al., 2022; Guo et al., 2023). The other aggregates neighbouring features implicitly, avoiding explicit alignment but leaving the network to learn the structure of rain motion entirely from data. Meanwhile, physical knowledge has mainly been introduced through rendering models, regularization terms, or rain synthesis (Yang et al., 2017; Hu et al., 2019), while geometric priors have been incorporated into attention biases (Liu et al., 2021; Sun et al., 2026). Although rain motion has been used for video rain synthesis (Wang et al., 2022), the motion structure of rain itself remains largely unused in guiding how information is aggregated across frames and scales.

Classical studies have shown that rain exhibits structured spatial and temporal properties rather than arbitrary corruption (Garg & Nayar, 2007; Barnum et al., 2010). Later work further characterized

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

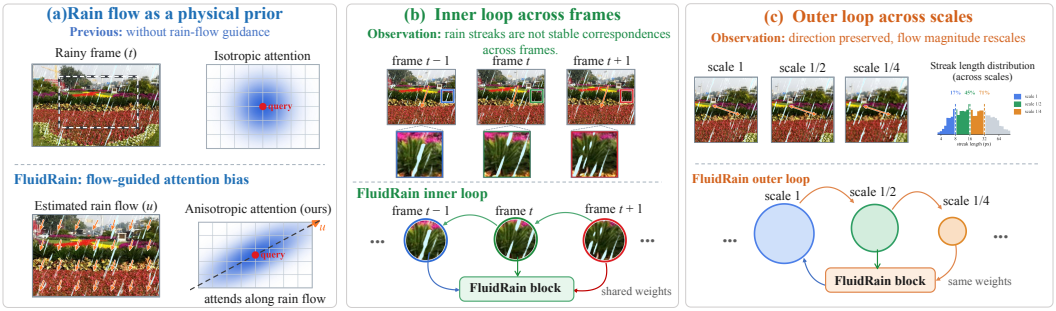

Figure 1: **Physical motivation and Loop-in-Loop design of FluidRain.** (a) Rain flow guides anisotropic attention. (b) Temporal persistence motivates the inner loop across frames. (c) Scale consistency motivates the outer loop across resolutions.

rain layers through attributes such as direction, scale, and temporal motion (Bossu et al., 2011; Yue et al., 2021). These regularities suggest that rain physics may provide not only guidance for feature aggregation, but also a principled basis for deciding where computation can be reused. At the same time, looped Transformers have recently regained attention in large language models (LLMs) as a parameter-efficient alternative to conventional depth scaling. Relaxed Recursive Transformers repeatedly reuse shared blocks across depth, while recent scaling-law analyses show that recurrence can provide measurable capacity gains despite parameter sharing (Bae et al., 2025; Schwethelm et al., 2026). These two lines of work suggest a natural connection between physical structure and recurrent computation. We therefore ask whether rain regularities can provide a principled basis for organizing parameter sharing across time and scale in video restoration.

To investigate this question, Figure 1 connects three rain regularities to the design of FluidRain. (a) Inspired by fluid mechanics, we model coherent rain motion within a frame as a divergence-free image-space rain-flow field, which guides anisotropic attention along rain streaks. (b) Across neighbouring frames, the flow remains persistent while individual streaks are renewed, motivating an *inner loop* that reuses the same rain-aware operator over complementary observations. (c) Across resolutions, rain-flow direction is preserved while streak extent rescales, motivating an *outer loop* that reuses the same operator across scales. We implement this idea as Loop-in-Loop, an alignment-free three-frame architecture with 0.80M parameters. Our main contributions are summarized below.

- We develop a fluid-mechanical rain-flow representation that guides attention along rain streaks, enabling alignment-free temporal aggregation with explicit rain-motion guidance.
- We use the temporal and scale regularities of rain flow to organize nested parameter sharing. The same rain-aware operator reads complementary neighbouring frames and is reused across three resolutions.
- We evaluate the design with controlled view and weight-sharing studies, RainSyn-Gust direction changes, real-rain diagnostics, and downstream perception. The model achieves a strong parameter-performance tradeoff with only 0.80M parameters.

2 Related Work

Video deraining. Early video deraining methods relied on temporal filtering, low-rank priors, and recurrent architectures to exploit redundancy across neighbouring frames (Kim et al., 2015; Liu et al., 2018; Yang et al., 2021; Yue et al., 2021). Rain-specific temporal structure has also been explored, with RDD-Net explicitly modeling inter-frame rain-streak motion (Wang et al., 2022). More recent methods increasingly aggregate temporal information through learned feature interactions, including attention-based architectures (Yang et al., 2023) and state-space models. Recent state-space models further improve long-range spatio-temporal modeling with more favorable computational complexity. RainMamba introduces locality-aware state-space scanning for video deraining (Wu et al., 2024), while VDMamba combines spatial and temporal state-space branches with adaptive multi-frame fusion (Sun et al., 2025). Another line of work explicitly establishes temporal correspondences through optical flow, deformable alignment, or alignment-and-fusion strategies (Chan et al., 2022; Xue et al., 2025). More recent methods have also explored semi-supervised learning,

test-time adaptation, and video diffusion models to improve generalization to real-world adverse weather (Lin et al., 2025). Overall, current video deraining methods increasingly emphasize effective temporal modeling, robust multi-frame fusion, and generalization beyond synthetic rain.

Physical priors and looped architectures. Physical modeling has long provided structured priors for rain removal. Classical studies describe rain streak formation through drop motion, exposure, and imaging geometry (Garg & Nayar, 2004; 2006; 2007). Learning-based methods later incorporated such knowledge through rain formation models, decomposition constraints, and learned rain synthesis (Yang et al., 2017; Hu et al., 2019; Yue et al., 2021). More recently, structured information has been injected directly into feature interactions. Relative-position biases encode spatial geometry in window attention (Liu et al., 2021), while motion-related quantities can guide spatio-temporal attention (Sun et al., 2026). Structure can also be imposed on how computation itself is reused. Recursive and looped networks repeatedly apply shared operators, increasing the effective processing depth without introducing a new set of parameters at every stage (Kim et al., 2016; Dehghani et al., 2018). This idea has recently received renewed attention in LLMs. Relaxed Recursive Transformers reuse shared Transformer blocks across depth (Bae et al., 2025), and recent scaling analyses further study the capacity gained from recurrence under parameter sharing (Schwethelm et al., 2026). Related forms of recursive reuse have also appeared in recent vision and image-restoration models (He et al., 2026). Existing looped architectures mainly organize such recurrence along network depth or iterative refinement.

3 METHOD

As shown in Figure 2, FluidRain models rain streaks as an image-space rain-flow field, projects the estimated flow onto the divergence-free subspace, and uses it to construct an anisotropic attention bias. Its scale and temporal regularities further enable the same rain-aware operator to be reused across resolutions and neighbouring frames, forming the Loop-in-Loop architecture for centre-frame restoration.

Notation. Given a three-frame rainy clip $\mathbf{I}_\tau \in \mathbb{R}^{3\times H\times W}$, $\tau \in \{t-1, t, t+1\}$, the network predicts the restored centre frame $\hat{\mathbf{I}}_t$, with clean target $\mathbf{I}_t^{\mathrm{gt}}$. On synthetic benchmarks, $\mathbf{I}_\tau = \mathbf{I}_\tau^{\mathrm{gt}} + \mathbf{R}_\tau$, where $\mathbf{R}_\tau$ is the rain layer. Upright bold upper-case symbols denote tensors, such as frames, feature maps and attention matrices. Upright bold lower-case symbols denote vector fields over the image domain Ω. Italic symbols denote scalars, and upright subscripts are descriptive labels.

3.1 PHYSICAL BACKGROUND AND RAIN-FLOW PROPERTIES

Rain flow. A rain streak is the image-space trace of a falling droplet integrated over the camera exposure. Let a droplet fall at terminal velocity v_t, be advected by wind $\mathbf{w}$, and be observed by a camera with focal length f and motion $\mathbf{c}$. For a droplet at depth $Z(\mathbf{x})$, its projected image velocity is

$$\mathbf{u}(\mathbf{x}) = \frac{f}{Z(\mathbf{x})}\left(v_t\hat{\mathbf{g}} + \mathbf{w} - \mathbf{c}\right)_\perp, \tag{1}$$

where $\hat{\mathbf{g}}$ denotes the direction of gravity and $(\cdot)_\perp$ extracts the image-plane component. Over an exposure of duration T, the corresponding streak is oriented along $\mathbf{u}$ and has approximate length $|\mathbf{u}|T$. We refer to $\mathbf{u}$ as the *rain flow*. It describes the rain structure within a frame and should not be confused with optical flow, which describes scene correspondence between frames. The following three properties of rain flow form the physical basis of FluidRain (proofs in Appendix A).

Proposition 1 (Divergence-free rain flow). Under the incompressible-flow assumptions, the image-space rain flow of one depth layer satisfies

$$\nabla \cdot \mathbf{u} = 0. \tag{2}$$

For an arbitrary estimate $\hat{\mathbf{u}}$, its nearest admissible divergence-free field in L_2 is

$$\Pi(\hat{\mathbf{u}}) = \hat{\mathbf{u}} - \nabla\phi, \qquad \nabla^2\phi = \nabla\cdot\hat{\mathbf{u}}. \tag{3}$$

For general camera motion and depth variation, this equality need not hold exactly. In that case, Π is used as a structured physical prior rather than a hard claim about the complete image motion.

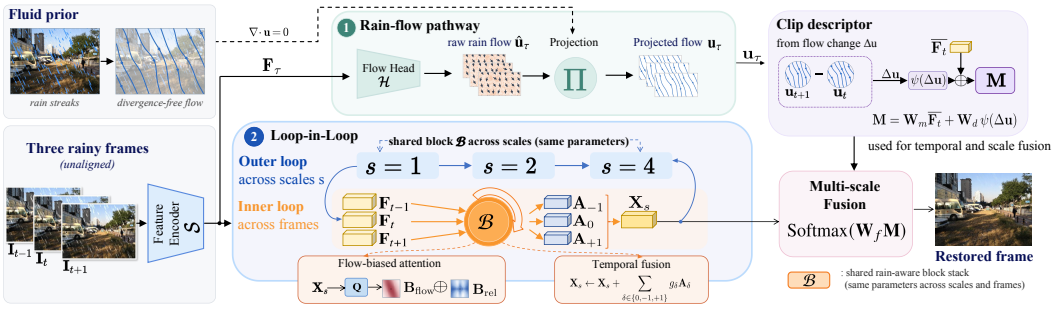

Figure 2: **Overview of FluidRain.** Divergence-free rain flow guides anisotropic attention, while Loop-in-Loop reuses the same operator across resolutions and neighbouring frames to aggregate complementary observations for centre-frame restoration.

Proposition 2 (Depth–scale equivariance). Let D_s denote downsampling by a factor s. For fixed projected droplet velocity, Eq. (1) gives

$$\mathbf{u}_{sZ} = \frac{\mathbf{u}_Z}{s} = D_s[\mathbf{u}_Z] \qquad \text{in grid units.} \tag{4}$$

Hence, changing depth or image resolution rescales the apparent streak extent while preserving its orientation.

Proposition 3 (Short-term flow persistence and streak renewal). Over a short inter-frame interval, the underlying rain flow varies slowly, while rain occupancy at a fixed scene location changes across exposures. Let

$$\Delta\mathbf{u}_\tau = \mathbf{u}_{\tau+1} - \mathbf{u}_\tau. \tag{5}$$

Under the same assumptions, $\Delta\mathbf{u}_\tau$ remains bounded for ordinary rain motion, while rain-induced occlusions in neighbouring frames provide distinct observations of the same scene.

3.2 Rain-Flow Estimation and Attention Bias

Rain-flow estimation. The rain flow in Eq. (1) depends on wind, scene depth, and camera motion, which are unknown at test time. We therefore infer it directly from the observed rain streaks. Given the frame feature $\mathbf{F}_\tau = \mathcal{S}(\mathbf{I}_\tau)$ from the feature encoder $\mathcal{S}$, a four-layer convolutional head $\mathcal{H}$ predicts a raw rain-flow field.

$$\hat{\mathbf{u}}_\tau = \mathcal{H}(\mathbf{F}_\tau). \tag{6}$$

We predict $\hat{\mathbf{u}}_\tau$ at quarter resolution, which is sufficient for capturing the slowly varying rain-flow structure while keeping the estimator lightweight. The same head $\mathcal{H}$ is reused for all input frames, so the flow estimator adds only 6,243 parameters.

However, $\hat{\mathbf{u}}_\tau$ is produced by an unconstrained neural predictor and is not guaranteed to satisfy the divergence-free property in Proposition 1. We therefore apply the divergence-free projection as a structured prior on the raw estimate.

$$\mathbf{u}_\tau = \Pi(\hat{\mathbf{u}}_\tau). \tag{7}$$

After projection, $\mathbf{u}_\tau$ serves as the rain-flow field for frame $\mathbf{I}_\tau$. Its direction indicates the local rain-streak orientation, while its magnitude sets the along-streak extent of the attention bias. We next use these two quantities to construct the rain-flow attention bias.

Rain-flow attention bias. Since rain flow characterizes rain-streak geometry rather than scene correspondence, we do not use it for feature warping. Instead, we convert its orientation and extent into an additive bias on the attention logits, encouraging attention to follow the local rain structure.

Consider a $w \times w$ attention window at scale s. Let $\mathbf{u}_w$ denote the mean rain flow over the window. The local rain flow $\mathbf{u}_w$ determines the streak orientation and the range of the attention bias. At scale s, we define

$$\hat{\mathbf{t}} = \frac{\mathbf{u}_w}{|\mathbf{u}_w|}, \qquad \hat{\mathbf{n}} \perp \hat{\mathbf{t}}, \qquad \ell_s = \max\left(\frac{|\mathbf{u}_w|}{s^2}, 1\right), \tag{8}$$

where $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$ denote the directions along and across the streak, respectively, and ℓ_s controls the attention-bias range. The additional $1/s$ keeps this range local across scales, while the rescaled flow direction remains unchanged.

These quantities directly parameterize the attention bias, in which $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$ specify the along-streak and across-streak directions and ℓ_s sets the effective range along the streak. For a query at $\mathbf{q}$ and a key at $\mathbf{k}$, let $\mathbf{d} = \mathbf{k} - \mathbf{q}$. The additive rain-flow bias $\mathbf{B}^{(h)}_{\text{flow}}$ for attention head h has entries

$$\left[\mathbf{B}^{(h)}_{\text{flow}}\right]_{\mathbf{d}} = -\frac{4}{w^2}\left[a_h(\mathbf{d}\cdot\hat{\mathbf{n}})^2 + b_h\frac{(\mathbf{d}\cdot\hat{\mathbf{t}})^2}{\ell_s^2}\right], \qquad a_h, b_h \geq 0. \tag{9}$$

The two terms control the bias in the directions normal and tangent to the rain streak, respectively. Displacements across the streak are penalized directly, whereas displacements along the streak are normalized by ℓ_s^2. Consequently, a larger ℓ_s yields a broader attention range along the rain direction, producing an anisotropic bias aligned with the local streak. The bias adds only two learnable parameters, a_h and b_h, for each attention head.

3.3 Loop-in-Loop across Scale and Time

A shared operator is most useful when each reuse receives complementary information. Rain provides compatible structure across both scale and time. The rain-flow direction is stable across scale, while neighbouring frames provide different observations under a similar short-term flow. We therefore reuse a shared operator across scales and temporal reads, forming the Loop-in-Loop design.

Outer loop across scales. Proposition 2 shows that downsampling preserves the rain-flow direction and rescales its magnitude by $1/s$. Therefore, after expressing the flow in the current grid as $\mathbf{u}/s$, the same rain-aware operator can be applied at different resolutions. Sharing is motivated primarily by this scale-consistent direction. The extent ℓ_s of Eq. (8) remains local at each scale. We thus reuse the same block stack $\mathcal{B}$ at three scales $s \in \{1, 2, 4\}$, i.e., at resolutions 1, $1/2$ and $1/4$.

For each scale s, the centre-frame state $\mathbf{X}$ (initialised with $\mathbf{F}_t$) is first resized as

$$\mathbf{X}_s = D_s(\mathbf{X}), \tag{10}$$

where D_s denotes resampling by a factor s. The rain flow of each frame τ is expressed on this grid as $\mathbf{u}_\tau/s$ and converted into the rain-flow bias using Eq. (9). The shared stack then updates the state,

$$\mathbf{X}_s \leftarrow \mathcal{B}(\mathbf{X}_s; \mathbf{B}_{\text{flow}}(\mathbf{u}_\tau/s)). \tag{11}$$

After upsampling by U_s, $U_s\mathbf{X}_s$ becomes the state $\mathbf{X}$ for the next scale. The three upsampled outputs are fused with weights $\mathrm{Softmax}(\mathbf{W}_f\mathbf{M})$, $\mathbf{M}$ being the clip descriptor defined below, and predict the residual added to $\mathbf{I}_t$.

Inner loop across frames. Proposition 3 shows that neighbouring frames share similar rain-flow structure while containing different rain-streak realizations, and thus provide complementary observations for restoring the centre frame. We therefore use the centre-frame state as the query and reuse the same attention operator to read keys and values from the centre and neighbouring frames without explicit alignment.

At scale s, let $\mathbf{X}_s$ denote the current centre-frame state. The centre state provides the query,

$$\mathbf{Q} = \mathbf{X}_s\mathbf{W}_Q, \tag{12}$$

while each temporal pass $\delta \in \{0, -1, +1\}$ obtains its keys and values from frame $t + \delta$.

$$\mathbf{K}_\delta = \mathbf{F}_{t+\delta}\mathbf{W}_K, \qquad \mathbf{V}_\delta = \mathbf{F}_{t+\delta}\mathbf{W}_V. \tag{13}$$

At each scale, the frame features are resized accordingly, and the centre pass uses the current state $\mathbf{X}_s$ instead of $\mathbf{F}_t$. The corresponding attention output is

$$\mathbf{A}_\delta = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}_\delta^\top}{\sqrt{d}} + \mathbf{B}_{\text{rel}} + \mathbf{B}_{\text{flow}}\left(\frac{\mathbf{u}_{t+\delta}}{s}\right)\right)\mathbf{V}_\delta, \tag{14}$$

where $\mathbf{B}_{\text{rel}}$ is the standard learned relative-position bias and $\mathbf{B}_{\text{flow}}$ is defined in Eq. (9). After reading the centre and two neighbouring frames with the shared attention operator, we obtain three attention outputs $\mathbf{A}_0$, $\mathbf{A}_{-1}$, and $\mathbf{A}_{+1}$. Rather than combining them equally, we predict three fusion weights from the centre-frame feature and a simple summary of the rain-flow change. Specifically, with $\Delta\mathbf{u} = \mathbf{u}_{t+1} - \mathbf{u}_t$, we define

$$\psi(\Delta\mathbf{u}) = \left[\sqrt{\overline{|\Delta\mathbf{u}|^2}},\ \overline{\cos\angle(\mathbf{u}_{t+1}, \mathbf{u}_t)}\right], \qquad \mathbf{M} = \mathbf{W}_m\overline{\mathbf{F}_t} + \mathbf{W}_d\,\psi(\Delta\mathbf{u}), \tag{15}$$

where $\mathbf{M}$ is a compact clip descriptor and ψ summarizes the frame-to-frame change in rain flow. Bars denote spatial means. The temporal fusion weights are then obtained as

$$g = \mathrm{Softmax}(\mathbf{W}_g \mathbf{M}), \qquad \mathbf{X}_s \leftarrow \mathbf{X}_s + \sum_{\delta \in \{0,-1,+1\}} g_\delta \mathbf{A}_\delta. \tag{16}$$

The fused feature is subsequently processed by the standard MLP sublayer, completing the inner-loop update. Together with the outer scale loop, this forms the Loop-in-Loop architecture, in which the same block stack is reused across resolutions and the same attention operator is reused to read different frames within each block.

3.4 Training

The network is trained primarily with an ℓ_1 reconstruction loss on the restored centre frame,

$$\mathcal{L}_{\mathrm{rec}} = \left\| \hat{\mathbf{I}}_t - \mathbf{I}_t^{\mathrm{gt}} \right\|_1. \tag{17}$$

On NTURain, the rain layer $\mathbf{R}_t = \mathbf{I}_t - \mathbf{I}_t^{\mathrm{gt}}$ also provides supervision for the rain-flow head. For each quarter-resolution cell containing rain, we extract a target streak orientation θ^* and length ℓ^*. Let θ and ℓ denote the corresponding quantities from the projected flow $\mathbf{u}_t$. The flow loss is

$$\mathcal{L}_{\mathrm{flow}} = \frac{1}{|\Omega_r|} \sum_{\Omega_r} \left[1 - \cos 2(\theta - \theta^*) + (\log \ell - \log \ell^*)^2 \right], \tag{18}$$

where Ω_r denotes cells containing visible rain. The total objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{flow}} \mathcal{L}_{\mathrm{flow}}. \tag{19}$$

We use $\lambda_{\mathrm{flow}} = 0.01$ on NTURain and $\lambda_{\mathrm{flow}} = 0$ on the three RainSyn benchmarks (Appendix C.2).

4 Experiments

4.1 Settings

Datasets. We evaluate on NTURain (Chen et al., 2018), RainSynLight and RainSynComplex (Liu et al., 2018), and RainSynAll100 (Yang et al., 2021), covering synthetic and real rain, light and heavy rain, non-linear background motion, and veiling effects. We further introduce **RainSyn-Gust** to evaluate robustness to changing rain motion.

Evaluation protocol. All methods are retrained under the same protocol on a single RTX 4090 GPU and evaluated on the full test sets. We report PSNR, SSIM and LPIPS on all four benchmarks, and FVD on NTURain. Appendix D.4 reports parameter efficiency and mobile runtime. For real rainy videos, we use the reference-free diagnostic F_{phys}. We compare with three groups of baselines, namely general image restoration (Liang et al., 2021; Zamir et al., 2022; Chen et al., 2022), single-image deraining (Xiao et al., 2022; Chen et al., 2023; 2024), and video deraining (Yue et al., 2021; Wang et al., 2022; Wu et al., 2024; Sun et al., 2025; 2026).

4.2 Comparison on four benchmarks

Table 1 compares FluidRain with representative image and video deraining methods. Under the unified 20k-iteration budget, FluidRain delivers competitive performance across all four datasets with only 0.80M parameters, and approaches much larger per-frame restoration models on NTURain, RainSynLight, and RainSynComplex. RainSynAll100 remains harder for the compact three-frame model, where Restormer and NeRD-Rain obtain higher PSNR. FluidRain also achieves competitive LPIPS among the evaluated video baselines. Figure 3 shows that FluidRain removes rain streaks while preserving scene structure and fine detail.

4.3 Analysis of the Loop-in-Loop Design

Table 2 examines where the benefit of Loop-in-Loop comes from. Repeating the centre frame gives only small and inconsistent gains. Changing the spatial view or reading a neighbouring frame improves restoration more clearly. The gain becomes larger when both immediate neighbours are used,

Table 1: **Quantitative comparison on video deraining benchmarks.** Best and second-best results are highlighted in bold and underline.

				NTURain			RainSynLight			RainSynComplex			RainSynAll100		
Method	Venue	#P (M)	Time (ms) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SwinIR-light	ICCVW'21	0.91	803	32.79	0.952	0.0572	33.25	0.956	0.0667	26.61	0.848	0.2232	32.49	0.924	0.1150
Restormer	CVPR'22	26.11	232	35.09	**0.962**	**0.0355**	37.19	**0.978**	**0.0241**	30.07	0.897	**0.1302**	38.06	**0.965**	**0.0417**
NAFNet	ECCV'22	29.16	**20**	33.88	0.956	0.0432	35.60	0.967	0.0443	28.04	0.854	0.1980	35.30	0.934	0.1007
IDT	TPAMI'23	16.42	249	33.23	0.952	0.0586	35.81	0.969	0.0404	27.97	0.860	0.1894	35.44	0.946	0.0733
DRSformer	CVPR'23	33.66	468	34.07	0.958	0.0474	37.46	0.977	0.0244	29.68	0.892	0.1356	37.41	0.958	0.0549
NeRD-Rain	CVPR'24	22.89	298	34.71	0.960	0.0444	37.47	**0.978**	0.0244	29.79	0.888	0.1404	**38.16**	0.959	0.0546
S2VD	CVPR'21	**0.53**	28	34.17	0.956	0.0504	25.36	0.913	0.1389	28.23	0.836	0.2242	31.53	0.878	0.1637
RDD-Net	ECCV'22	5.54	462	33.16	0.951	0.0603	36.89	0.971	0.0273	29.35	0.889	0.1304	33.86	0.929	0.1070
RainMamba	ACM MM'24	30.86	75	32.65	0.942	0.0527	33.49	0.934	0.0816	27.75	0.809	0.2861	29.89	0.864	0.1993
VDMamba	CVPR'25	12.70	168	34.00	0.953	0.0538	35.38	0.946	0.0946	27.85	0.806	0.2753	31.13	0.866	0.2047
DeLiVR	ICLR'26	5.63	920	32.79	0.944	0.0714	33.41	0.941	0.0948	26.47	0.785	0.3199	32.26	0.872	0.1937
FluidRain	–	0.80	271	**35.73**	0.961	0.0387	**37.52**	0.973	0.0316	**30.18**	**0.898**	0.1438	35.64	0.948	0.0869

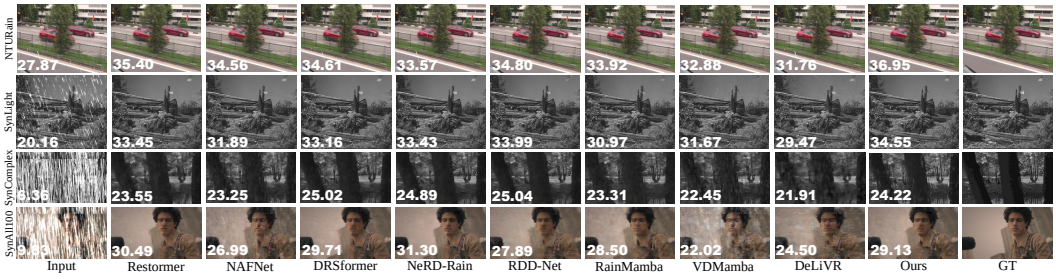

Figure 3: Qualitative comparison on video deraining benchmarks.

Table 2: **Analysis of the Loop-in-Loop design.** The notation below each setting shows the frame/view sequence. All variants except FluidRain omit the rain-flow pathway. * denotes the half-window-shifted tiling. *Untied scales* is *Neighbour + shift* with separate weights per scale.

	Baseline	Repeat	Complementary views					Temporal range		Weight Sharing	Full model
Setting	Centre Only	Repeat Centre	Shifted Centre	One neighbour	Neighbour + shift			Two neighbours	Five frames	Untied scales	FluidRain
Passes	t	t, t	t, t^*	$t, t{+}1$	$t, (t{+}1)^*$			$t, t{\pm}1$	$t, t{\pm}1, t{\pm}2$	$t, (t{+}1)^*$	$t, t{\pm}1$
#P (M)	0.79	0.79	0.79	0.79	0.79			0.79	0.79	2.01	0.80
NTURain	32.57	32.73	33.06	33.54	33.78			34.55	34.25	33.81	**35.73**
RainSynLight	33.29	33.17	33.79	35.65	35.66			36.85	36.71	36.11	**37.52**
RainSynComplex	26.56	26.61	27.47	27.85	27.96			28.78	29.04	28.55	**30.18**

showing that shared computation benefits from complementary observations rather than repetition alone. Extending the temporal range to five frames is not consistently helpful. Two neighbours already leave 0.98 of light-rain pixels clean in at least one frame, so farther frames add little there, while heavy rain keeps 0.30 occluded and is the only benchmark where five frames help. Untying the scale-specific operators raises the parameter count from 0.79M to 2.01M for only 0.03 to 0.59 dB gain. Reading a second neighbour with shared weights gives a larger 0.93–1.20 dB gain at no parameter cost. This supports the weight-tied three-frame setting $t-1, t, t+1$ as the Loop-in-Loop backbone. Adding the rain-flow pathway further improves this backbone on all three benchmarks, with the largest gain on RainSynComplex.

4.4 Analysis of the Rain-Flow Pathway

Table 3 evaluates the rain-flow pathway on top of the Loop-in-Loop backbone. With only 6.6K additional parameters, the full pathway improves PSNR by 0.67–1.40 dB. Removing the projection or the rain-flow bias degrades performance on all four benchmarks, while the differential descriptor gives a smaller gain. The pathway therefore benefits from both the projected field and its anisotropic bias.

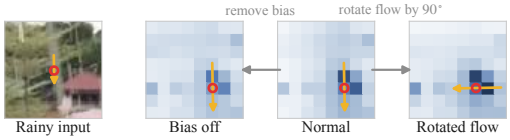

Figure 4: **Intervention on the rain-flow attention bias.** Removing or rotating the bias changes the attention direction.

Table 3: Ablation of the rain-flow pathway.

Variant	#P (M)	NTURain	RainSynLight	RainSynComplex	RainSynAll100
w/o rain-flow pathway	0.79	34.55	36.85	28.78	34.72
w/o projection	0.80	35.01	37.06	29.73	35.14
w/o rain-flow bias	0.80	35.07	36.96	29.61	35.08
w/o differential term	0.80	35.29	37.23	30.07	35.53
FluidRain (full)	0.80	**35.73**	**37.52**	**30.18**	**35.64**

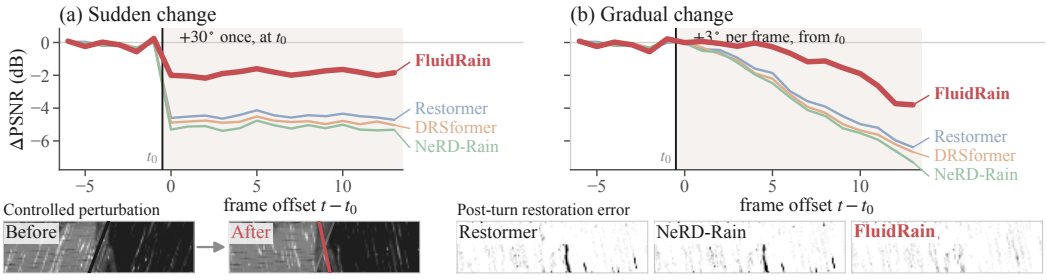

Figure 5: **Robustness to changing rain motion on RainSyn-Gust.** The bottom strip shows the controlled 30° perturbation and the corresponding post-turn restoration errors.

We jointly rotate the projected rain-flow fields of all three frames at inference. As the angular error increases, PSNR degrades overall (Table 4), while Figure 4 shows that the attention pattern rotates with the supplied flow direction. Removing the bias weakens this directional preference. Together, these results show that the rain-flow bias steers feature aggregation along the supplied rain direction.

Table 4: **Rain-flow direction intervention on NTURain.**

Offset	PSNR	Offset	PSNR
0°	35.73	60°	35.35
15°	35.69	75°	35.38
30°	35.58	90°	35.21
45°	35.50		

4.5 Relation to Explicit Alignment

As shown in Table 5, the RAFT optical-flow aligner (Teed & Deng, 2020) improves the Loop-in-Loop backbone by 0.54 dB on average, but provides no gain on RainSynComplex. After introducing rain-flow guidance, its average gain decreases to 0.18 dB. This reduced benefit motivates our alignment-free default design.

Table 5: **Relation to explicit alignment.** Each entry reports PSNR without / with RAFT.

Method	NTURain	RainSynLight	RainSynComplex	RainSynAll100
Loop-in-Loop (no flow)	34.55 / **35.62**	36.85 / **37.18**	**28.78** / 28.73	34.72 / **35.52**
FluidRain	35.73 / **36.04**	37.52 / **37.69**	**30.18** / 30.11	35.64 / **35.96**

4.6 Robustness to Changing Rain Motion

RainSynLight and RainSynComplex have nearly constant rain direction within each sequence. RainSyn-Gust introduces a sudden 30° step or a gradual 3°-per-frame direction change while keeping the clean background fixed. As shown in Figure 5, the three-frame Loop-in-Loop models show smaller drops than the per-frame baselines under the sudden change and remain stable under the gradual change. The flow-less backbone shows the same trend in Table 8, indicating that this robustness primarily comes from temporal reuse rather than the rain-flow bias itself. The descriptor $\psi(\Delta\mathbf{u})$ is trained on sequences with nearly constant direction and acts as a fusion weight in Table 3, so it is not expected to compensate an abrupt 30° turn.

4.7 Real-Rain Evaluation

Reference-free analysis. Real rainy videos lack clean references. We therefore use F_{phys} as a physics-inspired diagnostic of the removed residual. It balances whether the removed content is

physically consistent with rain (*precision*) and how much rain-like content is removed (*recall*).

$$\mathrm{Prec} = (TCP)^{1/3}, \qquad \mathrm{Rec} = \max\left(0,\, 1 - \frac{E_{\mathrm{rain}}(\hat{\mathbf{I}})}{E_{\mathrm{rain}}(\mathbf{I})}\right), \qquad F_{\mathrm{phys}} = \frac{2\,\mathrm{Prec}\,\mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}}.$$

Here, T measures temporal transience, C streak coherence, P consistency with the divergence-free rain-flow constraint, and $E_{\mathrm{rain}}(\cdot)$ rain-like residual energy (defined in Appendix E). As shown

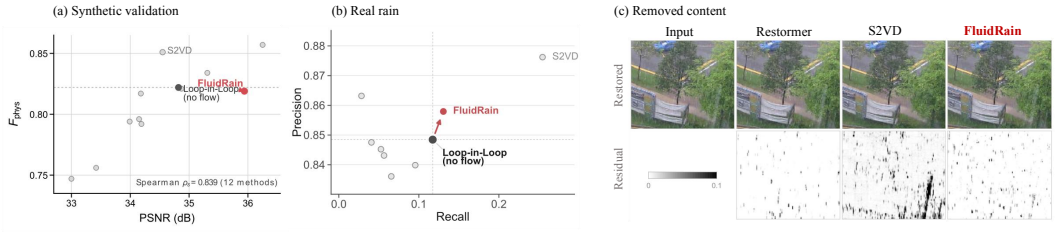

Figure 6: **Evaluation of F_{phys}.** (a) Correlation with PSNR on NTURain. (b) Precision–recall on real rain. (c) Restored crops and residuals.

in Figure 6, F_{phys} correlates with PSNR on synthetic NTURain (Spearman $\rho = 0.839$). On real NTURain, FluidRain improves precision/recall from 0.848/0.117 to 0.858/0.131 over the flow-less backbone. We use these scores only as diagnostics.

Evaluation on real rain. We further evaluate FluidRain on WeatherBench and our captured real-rain scenes without fine-tuning. Figure 7 shows qualitative results, while Appendix E.5 reports the quantitative comparison on WeatherBench.

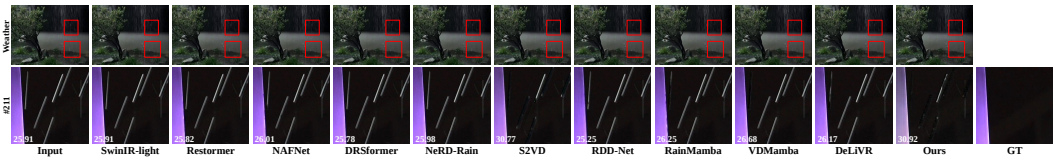

Figure 7: Qualitative results on real rain without fine-tuning.

4.8 Downstream Perception under Rain

We further evaluate the effect of deraining on downstream perception using KITTI Tracking (Geiger et al., 2012) and KITTI 2015 (Menze & Geiger, 2015) under synthetic light rain. FluidRain improves both detection and segmentation over the rainy inputs. In segmentation its gain approaches that of Restormer, a far larger restoration model, and in detection it exceeds DeLiVR and VDMamba. Figure 8 shows one detection and one segmentation frame, and Appendix C.4 reports the full results.



Figure 8: Downstream perception under synthetic light rain.

5 Conclusion

We presented FluidRain, a parameter-efficient video deraining framework that uses rain physics to shape both feature aggregation and parameter reuse. A projected rain-flow field defines anisotropic attention geometry, while Loop-in-Loop reuses the same rain-aware operator across neighbouring frames and resolutions. The resulting three-frame model uses only 0.80M parameters and remains competitive with substantially larger restoration networks. Controlled view studies show that complementary temporal observations are more valuable than repeated computation, and RainSyn-Gust confirms that the temporal loop remains stable when rain direction changes.

Reproducibility Statement

The physical assumptions behind the rain-flow prior are stated in Section 3.1 and derived in Appendix A. The complete Loop-in-Loop operator and the rain-flow pathway are specified in Appendix B, with pseudo code in Algorithm 1. Datasets, splits and data provenance are listed in Appendix C.1, the training and evaluation protocol and every hyper-parameter in Appendix C.2, the retraining protocol of the baselines in Appendix C.3 and the latency measurement in Appendix D.4 the downstream protocol in Appendix C.4, the RainSyn-Gust construction in Appendix D.3, and the definition of F_{phys} in Appendix E.1. Code, configuration files, trained checkpoints and the scripts that generate every table and figure of this paper will be released with the paper.

AI Use Statement

We used generative AI tools under our direction throughout this work. They also assisted us in writing the mathematical derivations and proofs in Appendix A, and in drafting and editing the manuscript. All experiments were run by us, and no experimental result, table, or figure was generated by an AI tool. We reviewed every AI-assisted derivation, checked every proof step independently, tested all AI-assisted code, and revised all AI-assisted text. We take full responsibility for the content of this paper.

References

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise LoRA. In *ICLR*, pp. 34282–34327, 2025.

Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *IJCV*, 86(2):256–274, 2010.

Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. The Helmholtz–Hodge decomposition—a survey. *IEEE TVCG*, 19(8):1386–1404, 2012.

Jérémie Bossu, Nicolas Hautiere, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *IJCV*, 93(3):348–367, 2011.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, pp. 5962–5971, 2022.

Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. Robust video content alignment and compensation for rain removal in a CNN framework. In *CVPR*, pp. 6286–6295, 2018.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, pp. 17–33, 2022.

Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *CVPR*, pp. 5896–5905, 2023.

Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *CVPR*, pp. 25627–25636, 2024.

Alexandre Joel Chorin. Numerical solution of the Navier–Stokes equations. *Math. Comput.*, 22 (104):745–762, 1968.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *CVPR*, 2004.

Kshitiz Garg and Shree K Nayar. Photorealistic rendering of rain streaks. *ACM TOG*, 25(3):996–1002, 2006.

Kshitiz Garg and Shree K Nayar. Vision and rain. *IJCV*, 75(1):3–27, 2007.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, pp. 3354–3361, 2012.

Qiyuan Guan, Qianfeng Yang, Xiang Chen, Tianyu Song, Guiyue Jin, and Jiyu Jin. WeatherBench: A real-world benchmark dataset for all-in-one adverse weather image restoration. In *ACM MM*, pp. 12607–12613, 2025.

Ross Gunn and Gilbert D Kinzer. The terminal velocity of fall for water droplets in stagnant air. *J. Meteorol.*, 6(4):243–248, 1949.

Yun Guo, Xueyao Xiao, Yi Chang, Shumin Deng, and Luxin Yan. From sky to the ground: A large-scale benchmark and simple baseline towards real rain removal. In *ICCV*, pp. 12063–12073, 2023.

Yunhong He, Zhengqing Yuan, Weixiang Sun, Yiyang Li, Yixin Liu, Yanfang Ye, and Lichao Sun. Vision-MoR: Scaling vision transformer via patch-level mixture-of-recursions. In *AAAI*, volume 40, pp. 4699–4707, 2026.

Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *CVPR*, pp. 8014–8023, 2019.

Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE TIP*, 24(9):2658–2670, 2015.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pp. 1637–1645, 2016.

Tae Hyun Kim, Seungjun Nah, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Conference on Computer Vision and Pattern Recognition*, pp. 1–21. IEEE, 2017.

John Frank Charles Kingman. *Poisson processes*, volume 3. Clarendon Press, 1992.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using Swin transformer. In *ICCVW*, pp. 1833–1844, 2021.

Chih-Hao Lin, Zian Wang, Ruofan Liang, Yuxuan Zhang, Sanja Fidler, Shenlong Wang, and Zan Gojcic. Controllable weather synthesis and removal with video diffusion models. In *ICCV*, pp. 13580–13591, 2025.

Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? Deep joint recurrent rain removal and reconstruction in videos. In *CVPR*, pp. 3233–3242, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 9992–10002, 2021.

Hugh Christopher Longuet-Higgins and Kvetoslav Prazdny. The interpretation of a moving retinal image. *Proc. R. Soc. Lond. B*, pp. 385–397, 1980.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, pp. 3061–3070, 2015.

Kristian Schwethelm, Daniel Rueckert, and Georgios Kaissis. How much is one recurrence worth? Iso-depth scaling laws for looped language models. *arXiv preprint arXiv:2604.21106*, 2026.

Jos Stam. Stable fluids. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 779–786. 2023.

Shangquan Sun, Wenqi Ren, Juxiang Zhou, Shu Wang, Jianhou Gan, and Xiaochun Cao. Semi-supervised state-space model with dynamic stacking filter for real-world video deraining. In *CVPR*, pp. 26114–26124, 2025.

Shuning Sun, Jialang Lu, Xiang Chen, Jichao Wang, Dianjie Lu, Guijuan Zhang, Guangwei Gao, and Zhuoran Zheng. DeLiVR: Differential spatiotemporal Lie bias for efficient video deraining. In *ICLR*, pp. 74299–74316, 2026.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pp. 402–419, 2020.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Shuai Wang, Lei Zhu, Huazhu Fu, Jing Qin, Carola-Bibiane Schönlieb, Wei Feng, and Song Wang. Rethinking video rain streak removal: A new synthesis model and a deraining network with video rain prior. In *ECCV*, pp. 565–582, 2022.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

Hongtao Wu, Yijun Yang, Huihui Xu, Weiming Wang, Jinni Zhou, and Lei Zhu. RainMamba: Enhanced locality learning with state space models for video deraining. In *ACM MM*, pp. 7881–7890, 2024.

Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE TPAMI*, 45(11):12978–12995, 2022.

Xinwei Xue, Jia He, Long Ma, Xiangyu Meng, Wenlin Li, and Risheng Liu. ASF-Net: Robust video deraining via temporal alignment and online adaptive learning. *Pattern Recognit.*, 158:110973, 2025.

Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, pp. 1685–1694, 2017.

Wenhan Yang, Robby T Tan, Jiashi Feng, Shiqi Wang, Bin Cheng, and Jiaying Liu. Recurrent multi-frame deraining: Combining physics guidance and adversarial learning. *IEEE TPAMI*, 44 (11):8569–8586, 2021.

Yijun Yang, Angelica I Aviles-Rivero, Huazhu Fu, Ye Liu, Weiming Wang, and Lei Zhu. Video adverse-weather-component suppression network via weather messenger and adversarial backpropagation. In *ICCV*, pp. 13154–13164, 2023.

Zongsheng Yue, Jianwen Xie, Qian Zhao, and Deyu Meng. Semi-supervised video deraining with dynamical rain generator. In *CVPR*, pp. 642–652, 2021.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pp. 5728–5739, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pp. 586–595, 2018.

Appendix

A Theoretical Derivations **13**

A.1 Incompressible Rain Flow and Projection 14

A.2 Depth–Scale Equivariance 15

A.3 Temporal Persistence and Streak Renewal 16

B Architecture Details **18**

B.1 Complete Loop-in-Loop Operator 18

B.2 Rain-Flow Pathway and Supervision 18

C Experimental Protocol and Reproducibility **19**

C.1 Datasets and Splits 19

C.2 Training and Evaluation Protocol 19

C.3 Baselines 20

C.4 Downstream Perception Protocol 20

D Extended Ablations and Robustness **20**

D.1 Error Maps 20

D.2 Alignment Details 20

D.3 RainSyn-Gust Protocol and Results 20

D.4 Efficiency and Temporal Quality 22

D.5 Generalization to Deblurring and Dehazing 22

E F_{phys} and Real-Rain Evaluation **24**

E.1 Metric Definition 24

E.2 Calibration and Synthetic Validation 25

E.3 Real-Rain Evaluation 25

E.4 Metric Scope 25

E.5 Zero-shot Evaluation on WeatherBench 26

A Theoretical Derivations

This appendix gives the common physical setting and derives the three rain-flow properties introduced in Section 3.1. Unless stated otherwise, measured statistics use the first 40 training sequences of RainSynLight and RainSynComplex. Notation follows Section 3. We use $\mathbf{u}$ for the rain flow of Eq. (1) and $\hat{\mathbf{u}}$ for an estimate of it.

Physical background and assumptions. **Camera.** A pinhole camera with focal length f in pixels and principal point at the origin maps $\mathbf{P} = (X, Y, Z)$ to $\mathbf{x} = (f/Z)(X, Y)$, Z being the optical axis. If $\mathbf{P}$ moves with velocity $\mathbf{V}$ relative to the camera, differentiating gives the image velocity

$$\dot{\mathbf{x}} = \frac{f}{Z}\mathbf{V}_{\perp} - \frac{V_Z}{Z}\,\mathbf{x}, \qquad \mathbf{V}_{\perp} := (V_X, V_Y), \tag{20}$$

so Eq. (1) is the case $V_Z = 0$. The radial term, and the terms a camera rotation adds, are discussed in Remark 2.

Drops. A drop of radius a falls at the speed at which drag balances weight, $\frac{1}{2}\rho_a C_d \pi a^2 v_t^2 = \frac{4}{3}\pi a^3 \rho_w g$.

$$v_t = \sqrt{\frac{8\,a\,\rho_w\,g}{3\,\rho_a\,C_d}} \approx 207\sqrt{a}\ \text{ m/s} \qquad (\rho_w/\rho_a \approx 816,\ C_d \approx 0.5,\ a \text{ in metres}), \tag{21}$$

the square-root law $v_t \approx 200\sqrt{a}$ used by Garg & Nayar (2007). Gunn & Kinzer (1949) measured terminal velocities from 4.0 to 8.8 m/s for drops with diameters from 1 to 4 mm, consistent with the scale given by Eq. (21). Relative to a camera translating at velocity $\mathbf{c}$, a drop at terminal velocity in a wind $\mathbf{w}$ moves at

$$\mathbf{V} = v_t \hat{\mathbf{g}} + \mathbf{w} - \mathbf{c}, \tag{22}$$

with $\hat{\mathbf{g}}$ the unit vector of gravity. During an exposure T the drop traces a streak of length $|\mathbf{u}|T$ along $\mathbf{u}$ whose width is the image diameter $2fa/Z$ of the drop.

Photometry. The drop covers a pixel for a time $\tau < 4a/v_t < 1.18$ ms (Garg & Nayar, 2007, Eq. 29), far below a video exposure $T \approx 1/30$ s. A pixel with a linear response integrates irradiance over T. Let the background contribute E_b and the drop $\bar{E}_r$ while it covers the pixel. The recorded intensity is

$$I = \frac{1}{T}\Big[(T-\tau)\,E_b + \tau\,\bar{E}_r\Big] = E_b + \frac{\tau}{T}\big(\bar{E}_r - E_b\big), \tag{23}$$

their Eq. (17) divided by T. Because a drop refracts light from a wide field of view towards the camera, $\bar{E}_r > E_b$. The rain term is therefore non-negative, giving the additive model $\mathbf{I}_\tau = \mathbf{I}^{\mathrm{gt}}_\tau + \mathbf{R}_\tau$ used in Section 3. Along a streak, the layer decreases linearly with the background intensity with slope $-\tau/T$ (their Eq. 18).

Assumption 1 (Wind and camera)**.** The wind $\mathbf{w}$ is a C^1 incompressible field, $\nabla\cdot\mathbf{w} = 0$, and horizontal. Across the imaged depth layer it has no component along the optical axis and no gradient of that component along the axis. Thus $w_Z = 0$ and $\partial_Z w_Z = 0$. The optical axis is horizontal, so $\hat{\mathbf{g}}$ lies in the image plane. The camera translates at velocity $\mathbf{c}$ with $c_Z = 0$ and does not rotate. A rotation about the optical axis is admissible (Remark 2).

Assumption 2 (Drops)**.** Drops of a given radius form a homogeneous Poisson point process in space with intensity n, independently across radii, and all drops of one radius share the velocity $\mathbf{V}$ of Eq. (22) over the imaged region. This is the spatial model of Garg & Nayar (2007).

Assumption 3 (Drag)**.** A drop of mass $m = \frac{4}{3}\pi a^3 \rho_w$ and velocity $\mathbf{v}$ obeys the quadratic drag law

$$m\,\dot{\mathbf{v}} = m\mathbf{g} - k\,|\mathbf{v}-\mathbf{w}|\,(\mathbf{v}-\mathbf{w}), \qquad k = \tfrac{1}{2}\rho_a C_d \pi a^2, \tag{24}$$

with constant C_d and $\mathbf{g} = g\hat{\mathbf{g}}$. In a constant wind its steady state is $\mathbf{v} - \mathbf{w} = v_t\hat{\mathbf{g}}$ with $kv_t^2 = mg$, which is Eq. (21).

A.1 Incompressible Rain Flow and Projection

Lemma 1 (Divergence of the rain flow)**.** *For the drops of a depth layer $Z = \mathrm{const}$ moving with the velocity $\mathbf{V}$ of Eq.* (22)*, the image flow of Eq.* (20) *satisfies, on the layer,*

$$\nabla\cdot\mathbf{u} = -\,\partial_Z w_Z - \frac{2V_Z}{Z} - \frac{x\,\partial_X w_Z + y\,\partial_Y w_Z}{f}. \tag{25}$$

Proof. On the layer $\mathbf{P}(\mathbf{x}) = (xZ/f,\, yZ/f,\, Z)$, so $\partial_x = (Z/f)\partial_X$ and $\partial_y = (Z/f)\partial_Y$. Differentiating $\mathbf{u} = (f/Z)\mathbf{V}_\perp - (V_Z/Z)\mathbf{x}$ gives

$$\partial_x u_x = \partial_X V_X - \frac{V_Z}{Z} - \frac{x}{f}\,\partial_X V_Z, \qquad \partial_y u_y = \partial_Y V_Y - \frac{V_Z}{Z} - \frac{y}{f}\,\partial_Y V_Z.$$

Adding the two, using that only $\mathbf{w}$ varies in $\mathbf{V}$ and that $\partial_X w_X + \partial_Y w_Y = -\partial_Z w_Z$ by incompressibility, gives Eq. (25). □

Helmholtz–Hodge decomposition. On the periodic image domain Ω, gradient fields and divergence-free fields are L^2-orthogonal. There is no boundary term.

$$\int_\Omega \mathbf{v}\cdot\nabla\phi = -\int_\Omega \phi\,\nabla\cdot\mathbf{v} = 0 \qquad \text{whenever } \nabla\cdot\mathbf{v} = 0. \tag{26}$$

Hence every $\hat{\mathbf{u}} \in L^2(\Omega;\mathbb{R}^2)$ splits uniquely into a divergence-free part and a gradient,

$$\hat{\mathbf{u}} = \mathbf{u} + \nabla\phi, \qquad \nabla\cdot\mathbf{u} = 0, \qquad \nabla^2\phi = \nabla\cdot\hat{\mathbf{u}}, \tag{27}$$

Taking the divergence gives the Poisson equation in Eq. (3). By Eq. (26), $\mathbf{u}$ is the divergence-free field nearest to $\hat{\mathbf{u}}$ (Bhatia et al., 2012). In Fourier space $\nabla\cdot$ is $i\mathbf{k}\cdot$ and ∇^2 is $-|\mathbf{k}|^2$, so for $\mathbf{k} \neq 0$

$$\phi_{\mathbf{k}} = -\,\frac{i\,\mathbf{k}\cdot\hat{\mathbf{u}}_{\mathbf{k}}}{|\mathbf{k}|^2}, \qquad (\nabla\phi)_{\mathbf{k}} = \frac{\mathbf{k}\,(\mathbf{k}\cdot\hat{\mathbf{u}}_{\mathbf{k}})}{|\mathbf{k}|^2}. \tag{28}$$

Proposition 1 (Divergence-free rain flow)**.** *Under Assumption 1, the image flow of Eq.* (1) *of a depth layer satisfies* $\nabla\cdot\mathbf{u} = 0$ *on* Ω*. For any estimate* $\hat{\mathbf{u}} \in L^2(\Omega;\mathbb{R}^2)$*, the field* $\Pi(\hat{\mathbf{u}})$ *of Eq.* (3)*, in Fourier form*

$$\Pi(\hat{\mathbf{u}})_{\mathbf{k}} = \Big(\mathbf{I}_2 - \frac{\mathbf{k}\mathbf{k}^\top}{|\mathbf{k}|^2}\Big)\hat{\mathbf{u}}_{\mathbf{k}} \quad (\mathbf{k}\neq 0), \qquad \Pi(\hat{\mathbf{u}})_0 = \hat{\mathbf{u}}_0, \tag{29}$$

with $\mathbf{I}_2$ *the* 2×2 *identity, is the* L^2*-orthogonal projection of* $\hat{\mathbf{u}}$ *onto the divergence-free fields. It is linear, idempotent, self-adjoint, real for real input, and parameter-free.*

Proof. Assumption 1 gives $V_Z = 0$ on the layer because $c_Z = w_Z = \hat{g}_Z = 0$. Therefore Eq. (20) reduces to Eq. (1). In Eq. (25), the second term vanishes because $V_Z = 0$. The third vanishes because $w_Z \equiv 0$ on the layer plane has zero in-plane derivatives. The first vanishes because $\partial_Z w_Z = 0$. Eq. (29) is $\hat{\mathbf{u}}_{\mathbf{k}} - (\nabla\phi)_{\mathbf{k}}$ with Eq. (28), i.e. the divergence-free part $\mathbf{u}$ of Eq. (27), hence the orthogonal projection. Mode by mode, $\Pi_{\mathbf{k}} = \mathbf{I}_2 - \mathbf{k}\mathbf{k}^\top/|\mathbf{k}|^2$ is the real symmetric projector onto $\mathbf{k}^\perp$. Thus Π is linear, idempotent and self-adjoint. The identity $\Pi_{-\mathbf{k}} = \Pi_{\mathbf{k}}$ preserves the conjugate symmetry of a real field. The $\mathbf{k} = 0$ mode is the mean flow. It has no divergence and is left untouched. The same decomposition is the pressure step of projection methods for incompressible flow (Chorin, 1968; Stam, 2023). □

Remark 1 (Discrete projection). The implementation applies Eq. (29) on the $H\times W$ periodic quarter-resolution grid with DFT wavevectors and spectral derivatives $i\mathbf{k}$ (Appendix B.2). For an even side N the index $N/2$ is its own negative modulo N. No non-zero wavenumber assigned to it can satisfy $k(-m) = -k(m)$. The spectral divergence of a real field then acquires an imaginary part, and discarding it breaks idempotence. For odd-order spectral derivatives, the Nyquist wavenumber is set to zero. Then $\mathbf{k}(-\mathbf{m}) = -\mathbf{k}(\mathbf{m})$ holds for every index. The proof above applies directly. The discrete Π is therefore idempotent and symmetric up to floating-point precision.

In cells without visible rain, the flow head receives no direct target. The projection acts on the whole predicted field and constrains it to the divergence-free subspace. Across the 40 sequences, the median structure-tensor coherence of the dominant streak direction is 0.86 for light rain and 0.89 for heavy rain. This supports the coherent rain-flow structure described in Section 3.1.

Remark 2 (Camera motion and depth variation). Eq. (25) makes the effect of deviations from Assumption 1 explicit. Relative motion along the optical axis, $V_Z \neq 0$ (a camera driving forward, wind towards the camera), adds the radial field $-(V_Z/Z)\mathbf{x}$ of divergence $-2V_Z/Z$, whose divergent part Π removes. A camera rotating at angular velocity $\boldsymbol{\omega}$ adds to the image velocity of a point at depth Z the roll term and the pan–tilt term (Longuet-Higgins & Prazdny, 1980)

$$\omega_z\,(y,\,-x) \qquad \text{and} \qquad \Big(\frac{\omega_x\,xy}{f} - \omega_y\Big(f + \frac{x^2}{f}\Big),\ \omega_x\Big(f + \frac{y^2}{f}\Big) - \frac{\omega_y\,xy}{f}\Big).$$

The roll term is divergence-free, with curl $-2\omega_z$. A rotation about the optical axis therefore preserves the divergence-free property. To first order in $|\mathbf{x}|/f$, the pan–tilt term is the uniform shift $(-\omega_y f,\,\omega_x f)$. This term is divergence-free and is included in $\mathbf{c}$. Its second-order part has divergence $3(y\omega_x - x\omega_y)/f$, which is zero at the principal point. When the visible drops span a range of depths, they share the direction of $\mathbf{V}_\perp$ but not the magnitude $f|\mathbf{V}_\perp|/Z$. Thus the rain-flow direction is preserved across depth, consistent with the direction used by the rain-flow bias and the outer loop.

A.2 Depth–Scale Equivariance

Let D_s denote area downsampling by an integer factor s. A displacement of d fine pixels is d/s coarse pixels, so the flow of the downsampled rain, in coarse-grid units, is the block mean divided by s.

$$(D_s\mathbf{u})(\mathbf{x}') = \frac{1}{s}\cdot\frac{1}{s^2}\sum_{\boldsymbol{\delta}\in\{0,\dots,s-1\}^2}\mathbf{u}(s\mathbf{x}' + \boldsymbol{\delta}). \tag{30}$$

Proposition 2 (Depth–scale equivariance)**.** *Fix* $\mathbf{V}$ *and let* $\mathbf{u}_Z = (f/Z)\mathbf{V}_\perp$ *be the rain flow of drops at depth* Z*. Then, in grid units,*

$$\mathbf{u}_{sZ} = \frac{\mathbf{u}_Z}{s} = D_s[\mathbf{u}_Z]. \tag{31}$$

The streak length $|\mathbf{u}|T$ and width $2fa/Z$ scale in the same way, while the direction is unchanged. Thus changing depth or image resolution rescales the apparent streak extent while preserving its orientation.

Proof. $\mathbf{u}_Z \propto 1/Z$ at fixed direction gives $\mathbf{u}_{sZ} = \mathbf{u}_Z/s$. The field is uniform over the layer, so its block mean in Eq. (30) is itself and $D_s[\mathbf{u}_Z] = \mathbf{u}_Z/s$. Length and width are proportional to $1/Z$ and are divided by s on the coarse grid. Identical grid-unit inputs give identical outputs. □

Proposition 2 motivates the outer loop across scales through the scale-consistent rain-flow direction. On RainSynLight, streak lengths span 7–78 px between the 10th and 90th percentiles (Figure 1c). The outer loop uses the three scales $s \in \{1, 2, 4\}$.

A.3 Temporal Persistence and Streak Renewal

The temporal property in Section 3.1 has two parts. The first is short-term persistence of the rain flow. The second is renewal of individual streak instances across neighbouring frames. Let Δt be the frame interval and $T \leq \Delta t$ the exposure. Subscripts $\|$ and h denote components along and orthogonal to $\hat{\mathbf{g}}$. The symbol $\perp$ continues to denote image-plane projection.

Lemma 2 (Relaxation to the wind)**.** *Under Assumption 3, a drop near its steady slip $v_t\hat{\mathbf{g}}$ relaxes to a change of the wind with time constant $\tau_p = v_t/g$ in the horizontal components and $\tau_p/2$ in the vertical one. A step change $\Delta\mathbf{w} \perp \hat{\mathbf{g}}$ of the wind with $|\Delta\mathbf{w}| \ll v_t$ therefore changes the drop velocity by $(1 - e^{-t/\tau_p})\Delta\mathbf{w}$ after a time t.*

Proof. Let $\mathbf{w}$ be constant and write the slip as $\mathbf{r} = \mathbf{v} - \mathbf{w} = v_t\hat{\mathbf{g}} + \boldsymbol{\delta}$ with $|\boldsymbol{\delta}| \ll v_t$. Then $|\mathbf{r}|\mathbf{r} = v_t^2\hat{\mathbf{g}} + v_t\boldsymbol{\delta} + v_t(\hat{\mathbf{g}} \cdot \boldsymbol{\delta})\hat{\mathbf{g}} + O(|\boldsymbol{\delta}|^2)$, and inserting this into Eq. (24) with $\mathbf{g} = (k/m)v_t^2\hat{\mathbf{g}}$ and $kv_t/m = g/v_t$ gives the linearised dynamics

$$\dot{\boldsymbol{\delta}} = -\frac{g}{v_t}\Big[\boldsymbol{\delta}_h + 2\,\delta_\|\hat{\mathbf{g}}\Big], \qquad \text{i.e.} \qquad \dot{\boldsymbol{\delta}}_h = -\frac{\boldsymbol{\delta}_h}{\tau_p}, \quad \dot{\delta}_\| = -\frac{2\,\delta_\|}{\tau_p}, \quad \tau_p = \frac{v_t}{g}. \tag{32}$$

A step $\Delta\mathbf{w}$ sets $\boldsymbol{\delta}_h(0) = -\Delta\mathbf{w}$, the drop velocity being continuous, so $\boldsymbol{\delta}_h(t) = -e^{-t/\tau_p}\Delta\mathbf{w}$ and $\mathbf{v}_h = \mathbf{w}_h + \boldsymbol{\delta}_h$ has changed by $(1 - e^{-t/\tau_p})\Delta\mathbf{w}$. □

For $v_t = 6$ m/s, $\tau_p = 0.61$ s, so at 30 fps a drop acquires $1 - e^{-\Delta t/\tau_p} = 0.053$ of a small wind change per frame.

Lemma 3 (Streak renewal across neighbouring frames)**.** *Let $C(\mathbf{x})$ be the convex set of positions at which a drop covers pixel $\mathbf{x}$, and $\tau_c < 1.18$ ms the time a drop moving at $\mathbf{V}$ needs to cross it. The drops that mark $\mathbf{x}$ in frame n, exposed over $[t_n, t_n + T]$, are those whose position at a reference time lies in the tube*

$$S_n(\mathbf{x}) = \bigcup_{t \in [t_n,\, t_n + T]} \big(C(\mathbf{x}) - t\mathbf{V}\big), \qquad S_{n+1}(\mathbf{x}) = S_n(\mathbf{x}) - \Delta t\,\mathbf{V}. \tag{33}$$

If $\Delta t - T > \tau_c$, the tubes of all frames are pairwise disjoint translates of one another.

Proof. A drop at $\mathbf{P}$ is at $\mathbf{P} + t\mathbf{V}$ at time t and covers the pixel during frame n if and only if $\mathbf{P} + t\mathbf{V} \in C(\mathbf{x})$ for some t in the exposure, which is Eq. (33). The tube is convex, being a convex set plus a segment, with extent $|\mathbf{V}|(\tau_c + T)$ along $\mathbf{V}$, while consecutive tubes are shifted by $|\mathbf{V}|\Delta t$ along the same line. They are therefore disjoint when $\Delta t > T + \tau_c$. For a convex set A, $A \cap (A + \mathbf{d}) = \emptyset$ implies $A \cap (A + j\mathbf{d}) = \emptyset$ for every integer $j \geq 1$. Otherwise, a common point $\mathbf{a} = \mathbf{a}' + j\mathbf{d}$ would place the convex combination $\mathbf{a}' + \mathbf{d}$ in $A \cap (A + \mathbf{d})$. □

Proposition 3 (Short-term flow persistence and streak renewal)**.** *Under Assumptions 1–3, the following two properties hold.*

(i) The drop density is conserved between frames, $\partial_t n = 0$. If the wind changes by a step $\Delta\mathbf{w} \perp \hat{\mathbf{g}}$ while the camera velocity is unchanged and has no vertical component, the streak direction turns within one frame interval by an angle $\Delta\theta$ with

$$\sin\Delta\theta \;\leq\; \frac{g\,\Delta t\,|\Delta\mathbf{w}|}{v_t^3}\sqrt{v_t^2 + \tfrac{5}{4}|\Delta\mathbf{w}|^2} \;\approx\; \frac{\Delta t}{\tau_p}\cdot\frac{|\Delta\mathbf{w}|}{v_t} \quad \textit{for } |\Delta\mathbf{w}| \ll v_t. \tag{34}$$

(ii) If $\Delta t - T > \tau_c$, the events "pixel $\mathbf{x}$ is marked by rain in frame n" are independent across frames with a common probability $p = 1 - e^{-n|S|}$, and a pixel marked in the centre frame is clean in at least one of m distinct neighbouring frames with probability $1 - p^m$.

Proof. (i) Density. The density of the drops of one radius satisfies $\partial_t n + \nabla \cdot (n\mathbf{V}) = 0$ with $\nabla \cdot \mathbf{V} = \nabla \cdot \mathbf{w} = 0$, so a uniform density stays uniform.

(i) Direction. We bound the change $\Delta\mathbf{v}$ of the drop velocity over one frame without linearising. After the step the slip $\mathbf{r} = \mathbf{v} - \mathbf{w}$ obeys, by Eq. (24),

$$\dot{\mathbf{r}} = \mathbf{g} - \frac{k}{m}|\mathbf{r}|\,\mathbf{r}, \qquad \mathbf{r}(0) = v_t\hat{\mathbf{g}} - \Delta\mathbf{w}, \qquad |\mathbf{r}(0)|^2 = v_t^2 + |\Delta\mathbf{w}|^2 =: R^2.$$

Its horizontal part obeys $\dot{\mathbf{r}}_h = -(k/m)|\mathbf{r}|\,\mathbf{r}_h$, so $|\mathbf{r}_h|$ decreases from $|\Delta\mathbf{w}|$. Since $g\,r_\parallel \le g|\mathbf{r}| = (k/m)v_t^2|\mathbf{r}|$,

$$\frac{d}{dt}|\mathbf{r}|^2 = 2g\,r_\parallel - \frac{2k}{m}|\mathbf{r}|^3 \le \frac{2k}{m}|\mathbf{r}|\big(v_t^2 - |\mathbf{r}|^2\big),$$

which is negative whenever $|\mathbf{r}| > v_t$, so $|\mathbf{r}| \le R$ throughout. The vertical part $r_\parallel$ stays in $(0, v_t]$, because $\dot{r}_\parallel = (k/m)(v_t^2 - |\mathbf{r}|r_\parallel)$ is positive at $r_\parallel = 0$ and non-positive at $r_\parallel = v_t$. Moreover, $\eta = v_t - r_\parallel$ obeys $\dot{\eta} = (k/m)\big[(r_\parallel^2 - v_t^2) + r_\parallel(|\mathbf{r}| - r_\parallel)\big] \le (k/m)|\mathbf{r}_h|^2/2$, since $|\mathbf{r}| - r_\parallel = |\mathbf{r}_h|^2/(|\mathbf{r}| + r_\parallel)$. Integrating over the frame, with $k/m = g/v_t^2$,

$$|\Delta\mathbf{v}_h| \le \int_0^{\Delta t} \frac{k}{m}|\mathbf{r}||\mathbf{r}_h|\,dt \le \frac{g\,\Delta t}{v_t^2}\,R\,|\Delta\mathbf{w}|, \qquad |\Delta v_\parallel| = \eta(\Delta t) \le \frac{g\,\Delta t}{2\,v_t^2}\,|\Delta\mathbf{w}|^2. \tag{35}$$

The two parts project onto orthogonal image directions, $\hat{\mathbf{g}}$ lying in the image plane, so

$$|\Delta\mathbf{v}_\perp|^2 \le |\Delta\mathbf{v}_h|^2 + |\Delta v_\parallel|^2 \le \Big(\frac{g\,\Delta t\,|\Delta\mathbf{w}|}{v_t^2}\Big)^2\Big(v_t^2 + \tfrac{5}{4}|\Delta\mathbf{w}|^2\Big).$$

Let $\mathbf{p} = \mathbf{V}_\perp$ be the image-plane relative velocity before the step and $\mathbf{p}' = \mathbf{p} + \Delta\mathbf{v}_\perp$ after one frame. For planar vectors,

$$\sin\Delta\theta = \frac{|\mathbf{p}\times\mathbf{p}'|}{|\mathbf{p}|\,|\mathbf{p}'|} = \frac{|\mathbf{p}'\times\Delta\mathbf{v}_\perp|}{|\mathbf{p}|\,|\mathbf{p}'|} \le \frac{|\Delta\mathbf{v}_\perp|}{|\mathbf{p}|} \le \frac{|\Delta\mathbf{v}_\perp|}{v_t},$$

where $|\mathbf{p}| \ge \mathbf{p}\cdot\hat{\mathbf{g}} = v_t$ because the wind is horizontal and the camera has no vertical velocity. This gives Eq. (34). Its leading term for small gusts uses $\tau_p = v_t/g$.

(ii) By Lemma 3 the tubes of different frames are disjoint, and for a homogeneous Poisson process the counts in disjoint sets are independent Poisson variables with mean $n|S|$ (Kingman, 1992, Ch. 2). Therefore the marking events are independent with $p = 1 - e^{-n|S|}$. Drops of different radii have different tubes and form independent processes (Assumption 2). Independence across frames therefore remains, with $p = 1 - \exp\big(-\sum_a n_a|S^{(a)}|\big)$. Given a marked centre frame, all m neighbours are marked with probability p^m. □

At $v_t = 6$ m/s and 30 fps, Eq. (34) bounds the direction change from a 2 m/s gust by 1.1° per frame and from a 10 m/s gust by 11°. The linearised model of Lemma 2 gives 1.0° and 5°, respectively. The larger difference at 10 m/s follows from the nonlinear drag response. Camera motion can induce additional changes in the image-space rain flow.

The benchmark sequences exhibit nearly constant rain direction within each sequence. The variation is 0.05° for light rain and 0.03° for heavy rain, while directions range from -33° to $+34^\circ$ across sequences. RainSyn-Gust introduces controlled changes in rain-streak direction to evaluate robustness to changing rain motion (Appendix D.3).

Using a rain-layer threshold of 0.06, the occlusion rate is 0.137 for light rain and 0.485 for heavy rain. Conditioned on occlusion in the previous frame, the corresponding probabilities are 0.169 and 0.623. Among centre-frame rain pixels, the fraction that is clean in at least one of the two neighbours is 0.977 for light rain and 0.698 for heavy rain. The prediction $1 - p^2$ from Proposition 3 is 0.981 and 0.765, respectively. The deviation under heavy rain indicates residual temporal dependence beyond the idealized Poisson model. The normalized cross-correlation between adjacent rain layers

peaks at 0.068 for light rain and 0.082 for heavy rain. This statistic uses 90 frame pairs from the first 30 sequences of each corpus and searches displacements up to 150 pixels. The peak locations are scattered rather than concentrated at a single displacement. In both benchmarks, individual streak instances are renewed across frames while the rain-flow direction remains nearly unchanged within each sequence.

Inner loop. These observations support the inner loop of Loop-in-Loop. Neighbouring frames share similar rain-flow structure while containing different rain-streak realizations, allowing the same rain-aware operator to aggregate complementary observations. Table 2 evaluates this design.

B Architecture Details

B.1 Complete Loop-in-Loop Operator

Algorithm 1 summarizes the Loop-in-Loop forward pass in Section 3. The outer loop reuses the same block stack across three scales, while the inner loop reuses the same attention operator across the centre and neighbouring frames.

In Algorithm 1, D_s denotes area downsampling and U_s bilinear upsampling. LayerNorm is omitted for clarity. The feature encoder $\mathcal{S}$ contains two 3×3 convolutions ($3 \to 96 \to 96$) with GELU, and the tail contains two 3×3 convolutions ($96 \to 96 \to 3$). The shared stack $\mathcal{B}$ has eight blocks with width 96, six heads, 8×8 windows, and an MLP ratio of 2. Inputs are reflect-padded when needed and cropped back after restoration.

B.2 Rain-Flow Pathway and Supervision

The rain-flow head $\mathcal{H}$ operates on the encoded feature $\mathbf{F}_\tau$. It uses a 1×1 convolution ($96 \to 16$), two stride-2 3×3 convolutions ($16 \to 16$), and a final 1×1 convolution ($16 \to 3$), with GELU between layers. The first two output channels form the quarter-resolution field $\hat{\mathbf{u}}_\tau$. The remaining channel does not enter the Loop-in-Loop operator. The projection Π is implemented with one FFT pair on the periodic quarter-resolution grid.

Algorithm 1 FluidRain Loop-in-Loop forward pass.

Require: frames $\mathbf{I}_\tau$, $\tau \in \{t-1, t, t+1\}$

$\mathbf{F}_\tau \leftarrow \mathcal{S}(\mathbf{I}_\tau), \quad \mathbf{u}_\tau \leftarrow \Pi\big(\mathcal{H}(\mathbf{F}_\tau)\big)$ **for all** τ

$\mathbf{M} \leftarrow \mathbf{W}_m \overline{\mathbf{F}_t} + \mathbf{W}_d\, \psi(\mathbf{u}_{t+1} - \mathbf{u}_t)$

$\mathbf{X} \leftarrow \mathbf{F}_t$

for s **in** $(1, 2, 4)$ **do**

  $\mathbf{X}_s \leftarrow D_s\mathbf{X}, \quad \mathbf{N}_\delta \leftarrow D_s\mathbf{F}_{t+\delta}$ **for** $\delta \in \{-1, +1\}$

  for each block of $\mathcal{B}$ **do**

    for δ **in** $\{0, -1, +1\}$ **do**

      $\mathbf{Q} \leftarrow \mathbf{X}_s\mathbf{W}_Q$

      $\mathbf{K}_\delta, \mathbf{V}_\delta \leftarrow \begin{cases} \mathbf{X}_s\mathbf{W}_K,\ \mathbf{X}_s\mathbf{W}_V, & \delta = 0, \\ \mathbf{N}_\delta\mathbf{W}_K,\ \mathbf{N}_\delta\mathbf{W}_V, & \delta \neq 0 \end{cases}$

      $\mathbf{A}_\delta \leftarrow \mathrm{Softmax}\Big(\mathbf{Q}\mathbf{K}_\delta^\top/\sqrt{d} + \mathbf{B}_{\mathrm{rel}} + \mathbf{B}_{\mathrm{flow}}(\mathbf{u}_{t+\delta}/s)\Big)\,\mathbf{V}_\delta$

    end for

    $\mathbf{g} \leftarrow \mathrm{Softmax}(\mathbf{W}_g\mathbf{M})$

    $\mathbf{X}_s \leftarrow \mathbf{X}_s + \sum_\delta g_\delta \mathbf{A}_\delta$

    $\mathbf{X}_s \leftarrow \mathbf{X}_s + \mathrm{MLP}(\mathbf{X}_s)$

  end for

  $\mathbf{Y}_s \leftarrow U_s\mathbf{X}_s, \quad \mathbf{X} \leftarrow \mathbf{Y}_s$

end for

$\boldsymbol{\alpha} \leftarrow \mathrm{Softmax}(\mathbf{W}_f\mathbf{M})$

return $\mathbf{I}_t + \mathrm{Tail}(\sum_s \alpha_s \mathbf{Y}_s)$

Table 6: **Dataset splits used in the experiments.**

Dataset	Train seq.	Test seq.	Test windows	Frame size
NTURain	22	8 syn. + 7 real	1,650	640×480
RainSynLight	160	25	675	352×288
RainSynComplex	190	25	675	352×288
RainSynAll100	899	100	400	varied

Table 7: **NTURain sensitivity to** λ_{flow} for the full three-frame model.

λ_{flow}	0	0.01	0.03	0.1
PSNR	35.51	**35.73**	35.67	35.56

For each attention window, the projected field is averaged to obtain $\mathbf{u}_w$. Its direction and magnitude define the rain-flow bias in Eq. (9). The differential descriptor in Eq. (15) uses the root-mean-square magnitude and mean cosine similarity of $\Delta\mathbf{u} = \mathbf{u}_{t+1} - \mathbf{u}_t$.

The flow loss in Eq. (18) is evaluated on the projected centre-frame field at quarter resolution. The target direction θ^* is obtained from the dominant orientation of the rain-layer structure tensor. The target length ℓ^* is the median principal-axis length of streak components overlapping each cell. Cells with insufficient rain-layer mass are excluded from Ω_r.

Flow supervision is used only on NTURain. On RainSynLight, RainSynComplex, and RainSynAll100, the rain-flow head is learned through the reconstruction objective alone, as described in Section 3.4. Training hyperparameters and the evaluation protocol are given in Appendix C.2.

C Experimental Protocol and Reproducibility

C.1 Datasets and Splits

We use NTURain (Chen et al., 2018), RainSynLight and RainSynComplex (Liu et al., 2018), and RainSynAll100 (Yang et al., 2021). Table 6 summarizes the splits used throughout the paper. The NTURain real sequences have no clean reference and are evaluated with F_{phys} in Appendix E. RainSyn-Gust is constructed from the RainSynLight and RainSynComplex test sequences (Appendix D.3).

Data handling. We use the public releases of all four benchmarks. For RainSynAll100, test sequence IDs are excluded from the training split. RainSyn-Gust is generated from the corresponding test sequences without re-encoding the unchanged background frames.

C.2 Training and Evaluation Protocol

Training. Training samples are five-frame windows centred at t. The full FluidRain model uses $\{t-1, t, t+1\}$, while the five-frame control in Table 2 additionally uses $t \pm 2$. We use 128^2 crops, horizontal flipping, random temporal reversal, and batch size 4. All models are trained for $20{,}000$ iterations with AdamW (Loshchilov & Hutter, 2017). The initial learning rate is $2{\times}10^{-4}$ with $1{,}000$ warm-up iterations and cosine decay to 10^{-6}. Weight decay is 10^{-4}, gradient clipping is 1, and EMA is 0.999. Training uses fp16 on one RTX 4090. Ablations and controls use the same schedule. **Evaluation.** We evaluate the checkpoint from the final training iteration on the full test set. PSNR and SSIM (Wang et al., 2004) are computed on 8-bit RGB. LPIPS (Zhang et al., 2018) is reported on all four benchmarks and FVD (Unterthiner et al., 2018) on NTURain. FVD uses R3D-18 features from non-overlapping 16-frame clips and is compared only within this evaluation protocol. Image restoration baselines receive the centre frame. Video methods use the temporal input of their evaluated configuration. All restoration metrics are computed on the restored centre frame. The real-rain protocol for F_{phys} is given in Appendix E.

Flow-loss weight. We use $\lambda_{\text{flow}} = 0.01$ on NTURain. Table 7 reports a sensitivity analysis using the full three-frame model over $\lambda_{\text{flow}} \in \{0, 0.01, 0.03, 0.1\}$. The flow loss is disabled on RainSynLight, RainSynComplex, and RainSynAll100.

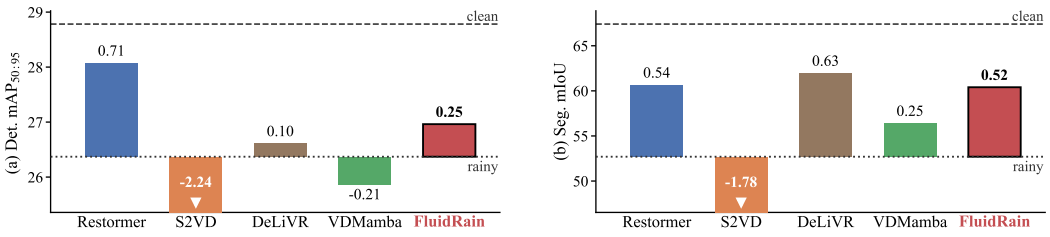

Figure 9: **Recovered fraction of the rain-induced loss** on the full test sets, for (a) detection $\text{mAP}_{50:95}$ and (b) segmentation mIoU. The dotted and dashed lines mark the rainy and clean inputs.

C.3 Baselines

We use the official implementations of all baselines and retrain them under the protocol above. Inputs are reflect-padded and cropped back when a model requires divisible spatial dimensions. IDT is evaluated with overlapping 128^2 tiles and stride 96, following its original evaluation procedure.

C.4 Downstream Perception Protocol

Data. Segmentation uses the 200 annotated frames of KITTI 2015 (Menze & Geiger, 2015), with each frame and its two neighbours forming a three-frame clip. Detection uses KITTI Tracking (Geiger et al., 2012), sampled every eight frames, yielding 953 clips and 5,071 boxes. We merge related object classes into car, truck, and person, and ignore boxes shorter than 25 px.

Rain synthesis. We add three consecutive RainSynLight rain layers to each clean clip. The rain sequence and crop are fixed per clip, and all methods receive the same rainy inputs.

Evaluation. Each method uses its RainSynLight checkpoint and restores the centre frame. Detection uses a frozen COCO-pretrained Faster R-CNN, while segmentation uses a frozen Cityscapes-pretrained SegFormer-B2. We report $\text{mAP}_{50:95}$, mIoU, and the recovered fraction $R = (M_{\text{restored}} - M_{\text{rainy}})/(M_{\text{clean}} - M_{\text{rainy}})$.

Results. Figure 9 reports the recovered fraction R of every method on the full test sets, and Figure 10 shows three more frames per task with all seven inputs.

D Extended Ablations and Robustness

D.1 Error Maps

Figure 11 complements the qualitative comparison in Figure 3 by showing the residual restoration errors on additional scenes.

D.2 Alignment Details

For the alignment control in Table 5, we use a frozen RAFT-small model to align each neighbouring frame to the centre frame before temporal aggregation. The aligner is kept fixed during training and evaluation. It introduces an additional motion-estimation network that runs for every neighbouring frame. No RAFT features are used by the rain-flow pathway.

D.3 RainSyn-Gust Protocol and Results

Construction. RainSyn-Gust is generated from the RainSynLight and RainSynComplex test sequences by rotating only the separable rain layer while keeping the clean background unchanged. Each sequence contains 31 frames and the change starts at $t_0 = 16$. G-cam applies a 30° step from t_0 onward. G-wind applies an additional 3° rotation at each frame from t_0 onward. All methods are evaluated without retraining.

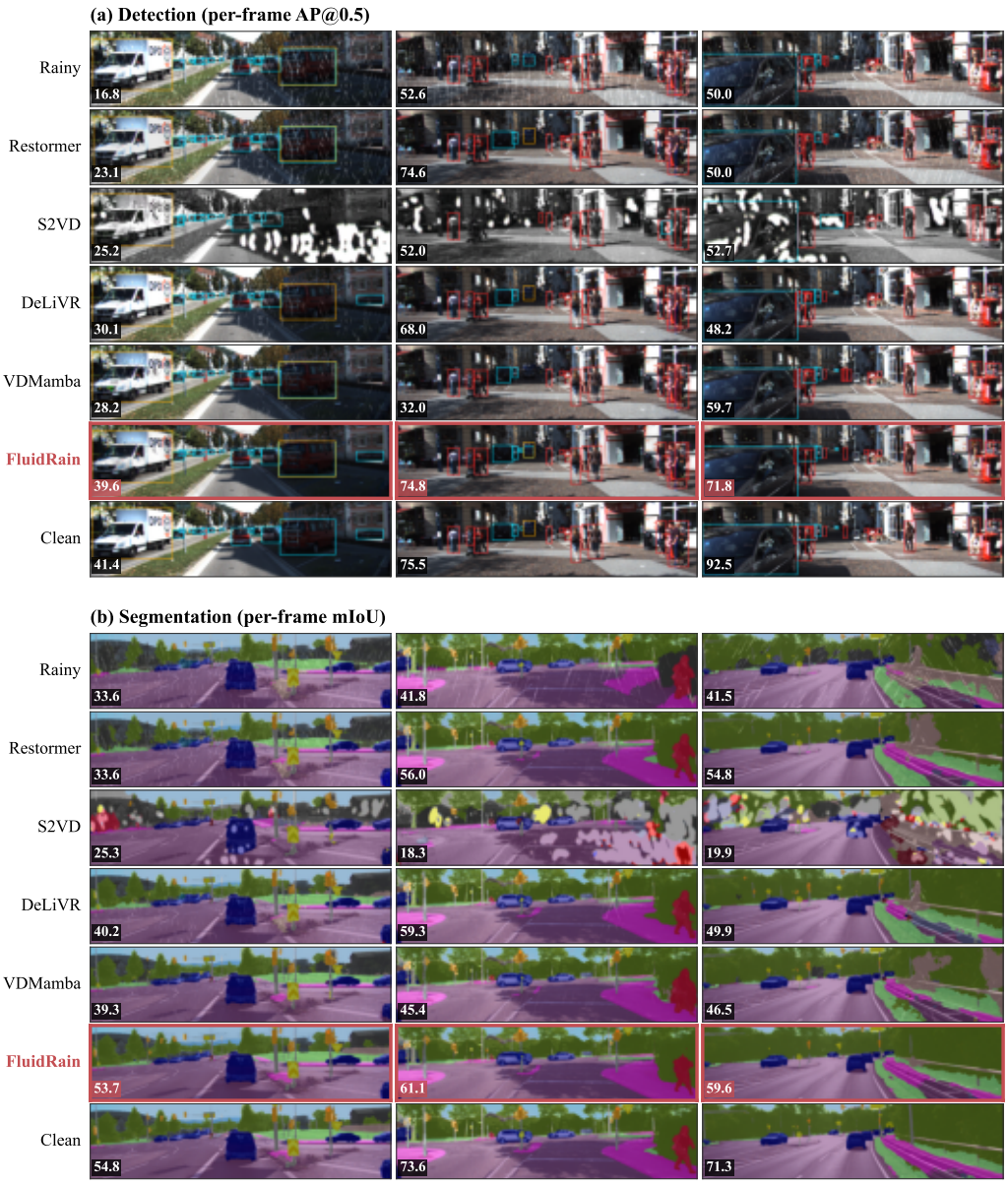

Figure 10: **More downstream examples under synthetic light rain.** (a) Detection and (b) segmentation, with per-frame AP@0.5 and mIoU ($\times 100$).

Evaluation. Each sequence provides 27 centre frames with a complete five-frame window. PSNR is measured on the centre 50% crop to exclude the empty border introduced by rotating the rain layer. For G-cam, we report the mean PSNR after t_0 and the PSNR drop at t_0 relative to the pre-change mean. For G-wind, we report the mean PSNR after t_0. The comparison focuses on restoration robustness under the controlled change in rain direction.

The RainSyn-Gust results support the robustness analysis in Section 4. Table 8 lists all methods, and Figure 12 shows one sequence around the 30° change. Under the sudden 30° change, the three-frame Loop-in-Loop models show smaller PSNR drops than all six per-frame baselines. Under the gradual change, they also maintain stable post-change performance across both rain conditions.

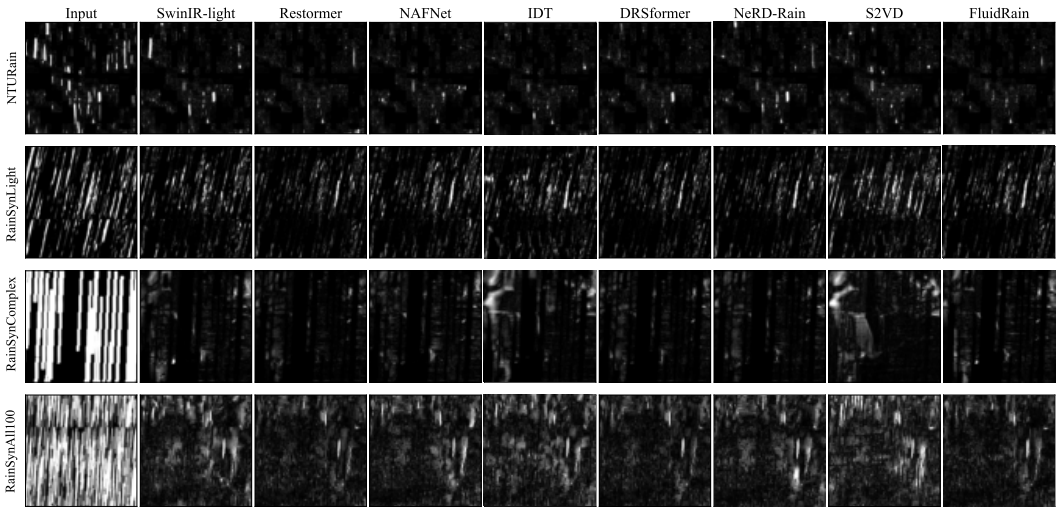

Figure 11: **Error maps** on one synthetic scene per benchmark, for the six image baselines, S2VD and FluidRain. Errors against the clean frame are amplified by $5\times$. The input column shows the corresponding rain layer.

Table 8: **RainSyn-Gust results.** Post-change PSNR is reported for G-cam and G-wind. G-cam also reports the PSNR drop at the 30° change.

	RainSynLight			RainSynComplex		
Method	G-cam Post	G-cam Drop	G-wind Post	G-cam Post	G-cam Drop	G-wind Post
SwinIR-light	30.59	2.38	31.30	22.71	3.69	23.81
Restormer	32.55	4.60	33.98	24.98	4.65	26.52
NAFNet	32.18	3.17	33.11	24.76	3.00	25.75
IDT	32.45	3.08	33.33	24.74	2.78	25.56
DRSformer	32.13	4.88	33.65	24.76	4.47	26.24
NeRD-Rain	31.91	5.31	33.56	24.13	5.23	25.87
S2VD	26.98	0.58	27.13	25.21	1.41	25.92
RDD-Net	35.59	4.94	37.08	24.63	5.73	26.77
RainMamba	31.25	1.25	31.72	22.60	2.71	23.94
VDMamba	32.68	2.55	33.68	22.70	1.50	23.40
DeLiVR	30.38	5.48	31.35	23.02	2.87	23.98
Loop-in-Loop (no flow)	34.70	1.68	35.28	24.77	2.14	25.80
FluidRain	34.53	2.00	35.18	24.75	2.62	25.90

D.4 Efficiency and Temporal Quality

Figure 13 compares PSNR against parameter count on four benchmarks. FluidRain uses only 0.80M parameters, 29–42$\times$ fewer than Restormer, NeRD-Rain, and DRSformer, while remaining competitive in PSNR. Only S2VD is smaller, with 4.9 dB lower average PSNR.

D.5 Generalization to Deblurring and Dehazing

FluidRain is designed for rain, but neither the Loop-in-Loop backbone nor the attention bias restricts the type of degradation. To test this, we fine-tune FluidRain on two other degradations without changing the architecture. For deblurring, we start from the NTURain checkpoint and train on three GoPro training sequences (Kim et al., 2017) at half resolution, then test on the first 20 frames of four GoPro test sequences. For dehazing, we start from the RainSynAll100 checkpoint and train on the haze-only frames that RainSynAll100 renders for the same scenes, then test on its 100 official test sequences. Neither set carries rain flow, so both runs use the L1 loss alone, for 2,000 and 3,000 iterations respectively. Dehazing raises PSNR from 16.28 to 19.88 dB and SSIM from 0.727 to 0.849. Deblurring gives a smaller gain, from 24.39 to 24.78 dB, which is expected from a short

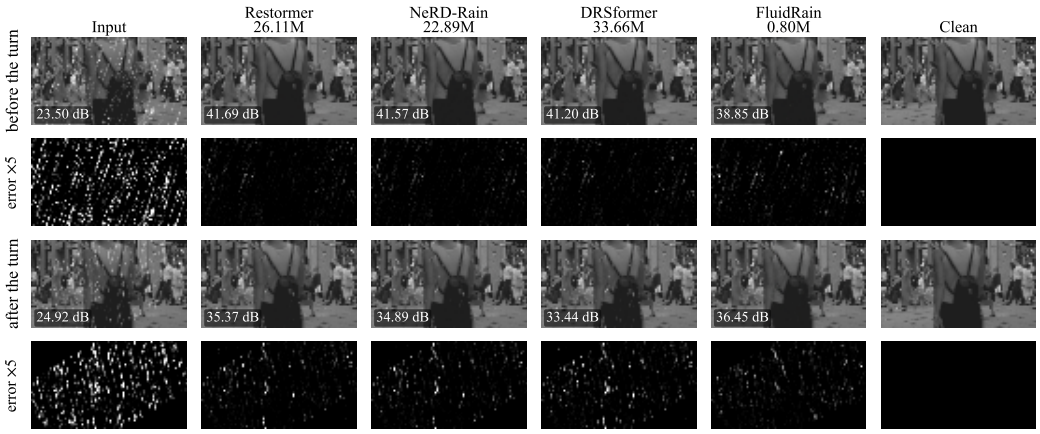

Figure 12: **RainSyn-Gust qualitative comparison** around the 30° G-cam change on RainSynLight. Error maps are amplified by $5\times$.

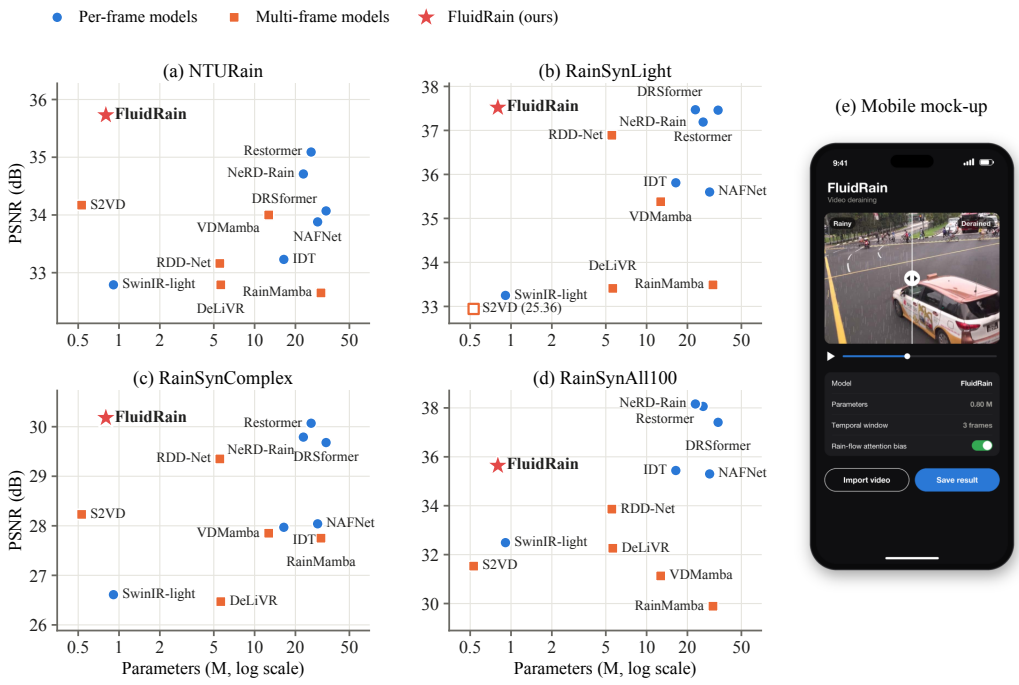

Figure 13: **Efficiency.** (a–d) PSNR versus parameter count on four benchmarks. (e) FluidRain running on a mobile device.

fine-tune on 282 frames. As shown in Figure 14, the fine-tuned model restores contrast and colour under haze and sharpens the edges of moving pedestrians under motion blur. The network therefore adapts to other degradations, although the rain-flow prior itself is specific to rain.



Figure 14: **Generalization to other degradations.** FluidRain fine-tuned for deblurring on GoPro (top, enlarged crop) and for dehazing on the haze-only frames of RainSynAll100 (bottom).

E F_{phys} AND REAL-RAIN EVALUATION

E.1 METRIC DEFINITION

F_{phys} evaluates a restoration from the rainy input $\mathbf{I}_t$ and restored frame $\hat{\mathbf{I}}_t$. We first convert both to grayscale and apply a high-pass filter by subtracting a Gaussian blur with $\sigma = 4$ px. The resulting input and output are denoted by $\mathbf{x}_t$ and $\mathbf{y}_t$, and the removed residual is $\mathbf{r}_t = \mathbf{x}_t - \mathbf{y}_t$. For metric computation only, consecutive input frames are aligned with a frozen RAFT model, and $\mathcal{W}$ denotes warping frame $t+1$ to frame t.

Precision. Precision measures whether the removed residual has the temporal and spatial properties of rain. It combines three terms.

Transience. Let ρ be the normalized cross-correlation between $\mathbf{r}_t$ and $\mathcal{W}(\mathbf{r}_{t+1})$, averaged over consecutive pairs. We define

$$T = 1 - \max(0, \rho). \tag{36}$$

Streak coherence. The residual is divided into 16×16 cells. In each cell, the structure tensor gives a dominant orientation ϕ, coherence c, and gradient energy e. We select the 30% highest-energy cells above an energy threshold of 2×10^{-4}. The coherence term is

$$C = \frac{\sum_{\Omega_{\text{sel}}} e\, c}{\sum_{\Omega_{\text{sel}}} e}. \tag{37}$$

Incompressibility. Let $\mathbf{v}$ be the unit streak-direction field on the selected cells, with signs resolved by the dominant frame direction. Using the projection Π from Eq. (3), we define

$$P = 1 - \frac{\|\mathbf{v} - \Pi\mathbf{v}\|^2}{\|\mathbf{v}\|^2}. \tag{38}$$

The precision score is

$$\text{Prec} = (TCP)^{1/3}. \tag{39}$$

It is set to zero when the removed residual has negligible energy.

Recall. Recall measures how much rain-like transient content remains after restoration. For each consecutive pair, we compute

$$\mathbf{d}_t^x = \mathbf{x}_t - \mathcal{W}(\mathbf{x}_{t+1}), \qquad \mathbf{d}_t^y = \mathbf{y}_t - \mathcal{W}(\mathbf{y}_{t+1}). \tag{40}$$

The structure tensor of each transient field provides (e, c, ϕ), and ω_t is the dominant rain direction estimated from the input. We then define

$$E_{\text{rain}}(\cdot) = \sum_t \sum_{\text{cells}} e\, c\, \cos^2(\phi - \omega_t), \tag{41}$$

Table 9: **Validation of F_{phys} on NTURain synthetic.** Spearman correlation is computed over the retrained methods and control variants. PSNR is computed on the same frames as F_{phys}, the first 80 frames of each synthetic sequence.

Method	PSNR	Precision	Recall	F_{phys}
Calibration				
Oracle (rain layer)	∞	0.904	0.845	0.869
Identity	29.06	0.000	0.000	0.000
Gaussian blur	24.91	0.454	0.881	0.596
Temporal median	24.65	0.564	0.725	0.632
Retrained methods and controls				
SwinIR-light	33.00	0.927	0.635	0.747
Restormer	35.31	0.926	0.765	0.834
NAFNet	34.18	0.926	0.737	0.817
IDT	33.42	0.929	0.645	0.756
DRSformer	34.19	0.932	0.693	0.792
NeRD-Rain	34.87	0.931	0.740	0.821
S2VD	34.55	0.954	0.775	0.851
Loop-in-Loop, two views (no flow)	33.99	0.935	0.696	0.794
FluidRain, two views	34.15	0.938	0.698	0.796
FluidRain, two views + RAFT	36.25	0.940	0.794	0.857
Loop-in-Loop (no flow)	34.82	0.934	0.740	0.822
FluidRain	35.94	0.935	0.734	0.819
Spearman with PSNR	—	—	—	0.839

Table 10: **Real-rain evaluation** on the seven NTURain real sequences.

Method	Precision ↑	Recall ↑
SwinIR-light	0.843	0.056
Restormer	0.836	0.066
NAFNet	0.840	0.095
IDT	0.845	0.053
DRSformer	0.848	0.041
NeRD-Rain	0.863	0.028
S2VD	0.876	0.255
Loop-in-Loop (no flow)	0.848	0.117
FluidRain	0.858	0.131

and

$$\text{Rec} = \max\left(0,\, 1 - \frac{E_{\text{rain}}(\hat{\mathbf{I}})}{E_{\text{rain}}(\mathbf{I})}\right). \tag{42}$$

The final score is the harmonic mean

$$F_{\text{phys}} = \frac{2\,\text{Prec}\,\text{Rec}}{\text{Prec} + \text{Rec}}. \tag{43}$$

E.2 Calibration and Synthetic Validation

We calibrate the metric on NTURain synthetic, where clean references are available. Every quantity is computed per sequence and averaged over sequences, using the first 80 frames of each of the 8 synthetic sequences and the first 60 frames of each of the 7 real sequences. The oracle removes exactly the rain layer, while identity, Gaussian blur, and temporal median provide reference points with different amounts and types of removed content.

Table 9 lists the scores. For calibration, we include both retrained baselines and control variants to cover a broader range of restoration quality. Across these evaluated models, F_{phys} has a Spearman correlation of 0.839 with PSNR. Figure 16a shows the same comparison. The oracle precision is lower than that of several derainers because the true rain layer also contains faint and overlapping streaks, so a higher precision reflects a more regular removed residual rather than a closer match to the rain layer.

E.3 Real-Rain Evaluation

On the seven real NTURain sequences, we report the precision and recall components of F_{phys} separately. Table 10 gives the quantitative results, and Figure 15 shows representative restorations.

Across the evaluated methods, precision ranges from 0.836 to 0.876, while recall ranges from 0.028 to 0.255. FluidRain reaches a precision of 0.858 and a recall of 0.131, compared with 0.848 and 0.117 for the Loop-in-Loop backbone without the rain-flow pathway.

E.4 Metric Scope

On RainSynComplex, F_{phys} spans 0.930–0.945 across the evaluated methods, with a Spearman correlation of 0.21 with PSNR. Figure 16b shows this saturation. NTURain synthetic is used for

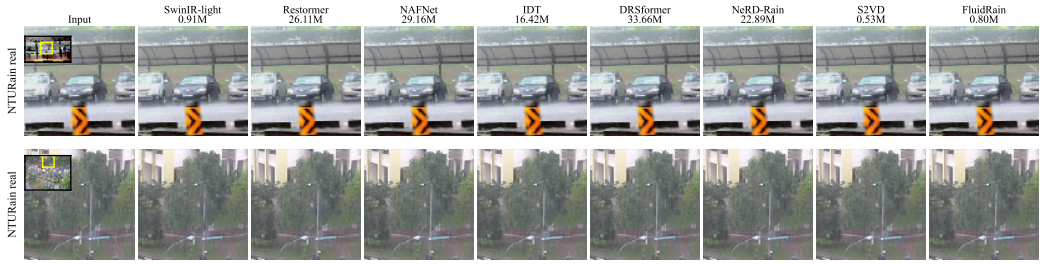

Figure 15: **Real-rain restorations** on two NTURain sequences without clean references.

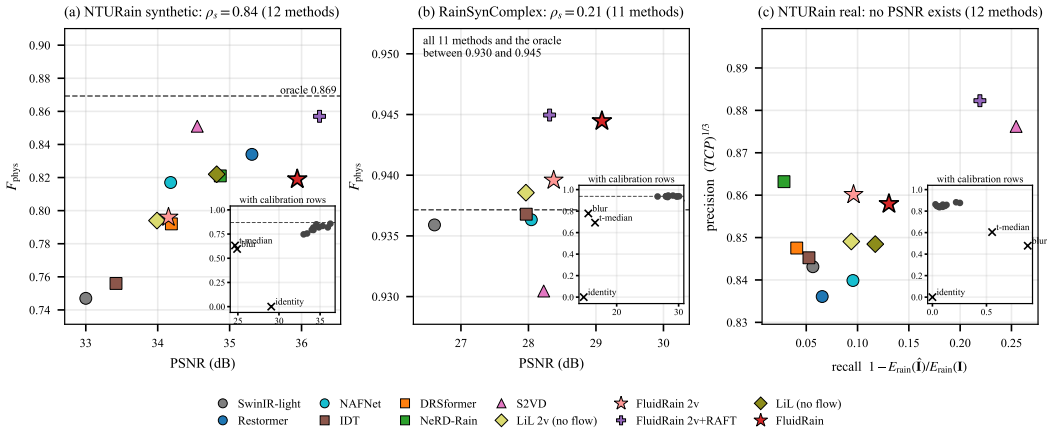

Figure 16: F_{phys} **evaluation.** (a) Comparison with PSNR on NTURain synthetic. (b) Saturation on RainSynComplex. (c) Precision and recall on NTURain real.

quantitative calibration. On real NTURain, precision and recall are reported separately. The metric's motion compensation relies on a frozen RAFT model. Where its flow errs under rain, background edges leak into the transient fields, so a method that also removes background detail can gain recall, as the Gaussian blur does (recall 0.881 in Table 9). Recall is therefore read together with precision, which penalizes such removals through T (precision 0.454 for the blur).

E.5 Zero-shot Evaluation on WeatherBench

We evaluate the NTURain checkpoints on WeatherBench (Guan et al., 2025) without fine-tuning. WB200 uses all 200 test images, while WB145 retains 145 images from bursts of at least three frames so that multi-frame models receive distinct neighbouring observations. PSNR, SSIM, and LPIPS are averaged over capture groups.

Table 11 shows that FluidRain remains competitive under zero-shot transfer. It achieves the best performance on WB200 and the second-best on WB145, where RainMamba performs best. Figure 17 provides representative qualitative results on real rain.

Table 11: **Zero-shot evaluation on WeatherBench.** NTURain checkpoints are evaluated without fine-tuning.

	WB200			WB145		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Rainy input	27.29	0.8569	0.3640	28.96	0.8551	0.3138
Per-frame models						
SwinIR-light	27.56	0.8660	0.3484	28.88	0.8547	0.3145
Restormer	27.96	0.8823	0.3293	28.91	0.8548	0.3148
NAFNet	27.37	0.8757	0.3332	28.94	0.8558	0.3137
IDT	27.62	0.8681	0.3448	28.93	0.8557	0.3133
DRSformer	27.92	0.8752	0.3313	28.94	0.8559	0.3131
NeRD-Rain	27.64	0.8697	0.3438	28.70	0.8536	0.3167
Multi-frame models						
S2VD	27.34	0.8576	0.3644	29.14	0.8588	0.3100
RDD-Net	27.47	0.8619	0.3540	28.96	0.8553	0.3138
RainMamba	27.72	0.8744	0.3355	**29.48**	**0.8693**	**0.3029**
VDMamba	27.76	0.8715	0.3365	29.13	0.8590	0.3098
DeLiVR	28.15	0.8778	0.3387	29.04	0.8583	0.3095
Loop-in-Loop (no flow)	27.86	0.8725	0.3379	29.05	0.8574	0.3112
FluidRain	**28.37**	**0.8823**	**0.3232**	29.27	0.8616	0.3074

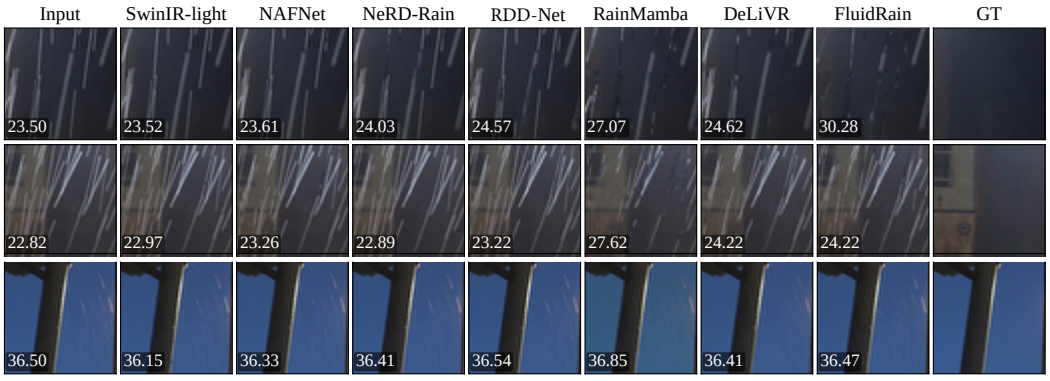

Figure 17: **Restorations on WeatherBench.** Representative heavy- and light-rain examples from WB145.