Exploring the Epipolar Consistency for Light Field Deraining

Chen Gao[1] **Youfang Lin**[1,2] **Wenbin Wang**[1] **Shuo Zhang**[1,2*]

[1]School of Computer Science & Technology, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, China
{gaochen,yflin,wangwenbin,zhangshuo}@bjtu.edu.cn

Abstract

Light Field (LF) deraining relies on leveraging redundant information between views to restore regions occluded by rains. Existing LF deraining methods typically rely on predicted depth maps to guide cross-view information interaction, while learning a brute-force mapping to separate rain from background content. However, the semi-transparent nature of rain streaks often leads to inaccurate depth estimation, which undermines geometric alignment and degrades restoration quality. More critically, the blending of rain and background during restoration makes brute-force mappings ineffective, resulting in visible rain artifacts in the output. In this paper, we first construct a consistency-constrained LF rain model by considering the consistency and the accumulation effects of rain layers. Based on this rain model, we synthesize a new dataset that includes various degradation levels. Then, we propose a novel **C**onsistency-**G**uided **R**ecovery Network (CGRNet) specifically designed for LF deraining. Inspired by the fact that background regions occluded by rain streaks often exhibit inconsistencies across all views, we introduce consistency maps to perceive degradation by aggregating pixel-wise consistency information along epipolar lines. Guided by consistency maps, the proposed modules adaptively sample and aggregate consistent features from clean background regions to restore fine details in degraded areas. Experiments validate our method's superiority over state-of-the-art approaches, and show that our dataset effectively enhances practical performance of current methods under real-world conditions.

1 Introduction

Capturing images in rainy conditions poses significant challenges, as rain streaks often obscure scene content and degrade overall image quality. These challenges are more pronounced in Light Field (LF) imaging since the rain streaks introduce inconsistent spatial degradations across views, resulting in disruption of inter-view geometric relationship. Such degradation significantly impair the performance of downstream tasks such as object detection [42] and autonomous driving [25].

The most straightforward approach to LF deraining is applying single-frame based methods [1, 40, 30, 3, 2] to each degraded view. Although these methods can produce visually appealing results for individual views based on rain features learned from training data, they inherently fail to preserve the intrinsic disparity constraints in LFs, meaning that the same point in the scene (assuming all surfaces are Lambertian) may exhibit inconsistent brightness across different views. To address the challenges associated with maintaining consistency across views, several LF deraining approaches have recently been proposed [8, 31, 32]. These methods generally utilize the estimated depth map to integrate complementary information on Epipolar Plane Images (EPIs) or 4D LFs for rain removal. However,

*Corresponding author.

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

they overlook two critical limitations under rainy conditions: (1) Accumulated rain layers disrupt epipolar consistency, leading to inaccurate depth estimation. These erroneous depth maps further cause geometric misalignment among views, degrading the final performance. (2) The blending of rain and background leads to strong entanglement during restoration. Simplified feature interaction strategies often fail to disentangle the two, leaving residual rain artifacts in the output.

In recent years, "seek-recover" paradigms have demonstrated remarkable effectiveness in deraining tasks. Inspired by these frameworks, we not only utilize epipolar consistency to capture complementary information but also explore how rain streaks explicitly break this consistency to localize the degradation. Under rain-free conditions, background regions exhibit Lambertian properties, leading to consistent appearances across views. However, when semi-transparent rain streaks overlap with the background, this Lambertian assumption breaks down, making it difficult to identify corresponding background content across all views. The regions exhibiting weak correspondence are more likely to be degraded by rain, whereas regions with strong correspondence can serve as reliable complementary information for restoration. Motivated by this observation, we use the epipolar consistency to characterize degradation in each region of the LFs and guide the recovery process.

Another challenge lies in the domain gap between synthetic and real-world data. Existing synthetic LF deraining datasets rely on single-image rain models without considering the intrinsic occlusion relationships and rain layer consistency of LFs. Models trained on them often generalize poorly to real-world scenarios. Consequently, bridging this gap is essential for robust practical performance.

In this paper, we first analyze the consistency and accumulation characteristics of rain layers in LFs and construct a consistency-constrained LF rain model. Based on this model, we propose a novel approach to synthesize a paired LF deraining dataset. Then, we propose a novel **C**onsistency-**G**uided **R**ecovery Network (CGRNet) for LF deraining. Our approach explores the consistency difference between degraded and non-degraded regions and introduces the consistency maps, where the consistency information of pixels along epipolar lines is aggregated to perceive degradation. Meanwhile, guided by the extracted consistency maps, our designed consistency-guided sparse EPI transformer and 4D refinement module directly sample and interact the consistent features from clean background regions to restore fine details in degraded areas. Experiments on synthetic and real-world scenes show that our method outperforms state-of-the-art LF deraining methods in quantitative and visual comparisons. Moreover, we compare our dataset with existing ones and show that it significantly enhances the performance of current methods under real-world conditions.

2 Related work

Single-frame-based Deraining. With the advancement of deep learning, a variety of architectures and strategies (e.g. multi-scale [15, 17], multi-stage [18, 39], and multi-branch [27, 41] approaches) have been explored to enhance the effectiveness of deraining. Recently, driven by the capability in modeling non-local information, transformer-based frameworks [9, 5] have emerged as powerful tools for rain removal [1, 40]. For example, Xiao et al. [30] proposed an image deraining transformer that incorporates spatial- and window-based self-attention modules. To make the model focus on clean background cues for restoration, some methods [3] attempt to 'seek' and recover background by modeling the degradation relationships of scene. However, the restoration of degraded background heavily relies on inpainting techniques, limiting their ability to restore complex scenes.

LF-based Deraining. Recently, redundancy between views of LF shows significant potential for recovering rain-degraded areas. Ding et al. [8] proposed a GAN-based architecture to remove rain streaks from 3D EPIs, which restores clear views by sequentially estimating depth maps and leveraging rain-background correlations. Nevertheless, it is restricted to exploiting views along the same row or column, failing to utilize the full angular information. To capture broader spatial-angular information, Yan et al. [31] employed 4D convolutional layers within an adversarial recurrent network to progressively eliminate rain streaks across views. Despite these advancements, a critical drawback of such methods is that rain layers inherently disrupt cross-view consistency, which often leads to inaccurate depth estimation and causes compounding errors in subsequent cross-view feature interactions. In addition, the disparity map of the clean background is required during training, which presents significant challenges in real scenes. To address the scarcity of real-world paired data for supervised training, Yan et al. [31] synthesize LF deraining datasets by directly applying single-image models to individual views. Due to the lack of consideration for rain consistency in LF imaging, the

gap between these synthetic datasets and real scenes is so large that current models must employ semi-supervised learning (e.g. GAN in [31]) to improve performance in real scenes.

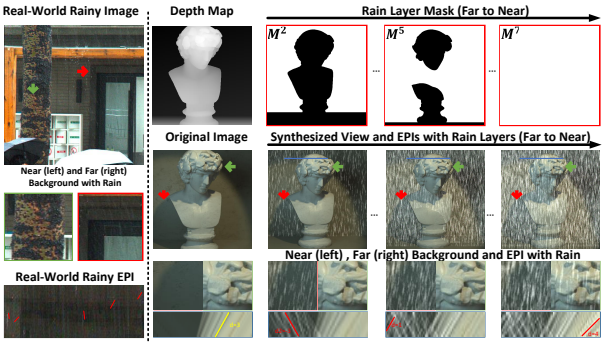

Figure 1: The LF in real scene alongside an example of our synthesis pipeline.

Table 1: Comparison among different de-raining datasets.

Dataset Characteristics	Single		LF	
	Rain100L	Rain100H	RLMB	Ours
Rain Appearance Diversity	✓	✓	✓	✓
Multi-layer Rain	×	✓	×	✓
Rain Accumulation	×	×	×	✓
Rain Consistency	-	-	×	✓
Spatially-Varying Rain	×	×	×	✓

3 Consistency-Constrained LF Rain Model

According to the two-plane model [16], a LF can be represented by a 4D tensor $\mathcal{L}(u, v, x, y) \in \mathbb{R}^{U \times V \times H \times W}$. $\mathbf{u} = (u, v)$ represents the angular coordinates and $\mathbf{x} = (x, y)$ represents spatial coordinates. The view at angular coordinate $\mathbf{u}$ can be denoted as $L_{\mathbf{u}} \in \mathbb{R}^{H \times W}$.

3.1 Rain Consistency Analyses

The rain layers in LFs are constrained by three factors:

- **Rain layer consistency:** According to traditional photometric consistency [37], rain streaks recorded across different views should satisfy the disparity constraint. Specifically, as illustrated in Fig. 1, corresponding points of a rain streak across views form straight lines (highlighted in red) in the EPI, reflecting their consistent geometric alignment.
- **Multi-layer rain accumulation:** In rainy conditions, multiple rain layers accumulate throughout the scene, resulting in varying degrees of degradation depending on the depth of the background. As shown in Fig. 1, areas closer to the camera (e.g., trees) are affected by fewer rain layers, while distant regions (e.g., windows) accumulate more layers. Due to the semi-transparency of rain streaks, this accumulation blurs details in deeper regions, and as more layers overlap, the resulting color tends to become increasingly white, further reducing background clarity.
- **Occlusion relationship:** In real-world rainy scenes, the epipolar lines of rain streaks in EPIs span the entire disparity range and may overlap with different background regions across views due to disparity differences between the rain layers and the background. Each rain layer respects realistic occlusion constraints, overlapping only with background content or deeper rain layers.

3.2 Model Construction

Based on the analysis above, we construct a consistency-constrained LF rain model following Physics-Based Rendering protocols [12, 35]. We first introduce the rain generation process for a single view in LFs and subsequently detail how to propagate it across all views. Physically, rain streaks exhibit a complex volumetric distribution in 3D space [14]. The front-layer rain streaks only blur the deeper rain streaks and the background behind them. Additionally, real-world rain exhibits inherent semi-transparency, allowing deeper scene details to remain partially visible [12]. To mathematically simulate these effects, our rain model is formulated as follows:

$$I = B + \alpha \cdot \sum_{d=d_{min}}^{d_{max}} M^d \cdot R^d, \tag{1}$$

where B represents the background, R^d denotes the rain layer at disparity d, and α is the overlay coefficient. d_{min} and d_{max} indicate the disparities corresponding to the minimum and maximum

depth of scene, respectively. Furthermore, M^d is the occlusion mask to constrain the degradation area affected by R^d. The example masks are shown in Fig. 1, where $M^d(\mathbf{x}) = 1$ labels the areas behind the related rain layer R^d. Using the mask ensures the rain layer only affects deeper scene information. As in synthesized view, areas closer to the camera (labeled as green arrow) have fewer rain layers and areas far away (labeled as red arrow) have more rain layers.

Since LF contains multiple views, the synthesized rain layers must satisfy disparity constraints across views. Specifically, once the rain layer for a particular view $R^d_{\mathbf{u}^*}$ is synthesized, the corresponding rain layer for another view $R^d_{\mathbf{u}}$ is calculated by shearing $R^d_{\mathbf{u}^*}$ as follow:

$$R^d_{\mathbf{u}}(\mathbf{x}) = R^d_{\mathbf{u}^*}(\mathbf{x} + (\mathbf{u} - \mathbf{u}^*) \cdot d). \tag{2}$$

Rather than independently calculating the rain layer for each view, we shear the rain layer from the central view to all other views according to the disparity, thus ensuring the disparity constraint is satisfied across the LF. The occlusion mask $M^d_{\mathbf{u}}$ is also obtained using the shearing operation in a similar way. Finally, we render all views in LF following Eq. 1. It can be clearly seen that the synthesized rains cross views follow the disparity constraint and form the lines in EPIs similar to the real-world EPIs (red lines) in Fig. 1. Based on this model, we also develop a consistency-constrained LF rain synthesis pipeline to generates rainy LF images.(Algorithm 1, detailed in Appendix A.)

Table 1 compares our rain model with existing approaches. Single-image datasets focus solely on rain appearance diversity (e.g., density, intensity, and length) without considering disparity constraints. Among LF dataset, RLMB [31] renders the LF rain layers by the particle system of Blender [6] and adds them linearly to the real-world LF images. On one hand, their approach applies motion blur independently to rain layers in each view, neglecting cross-view consistency of rain layers. On other hand, their method directly overlays rain layers onto the background, disregarding the depth relationship between the rain layers and the background. In contrast, we formulate the multi-layer rain accumulation over a defined depth range $[d_{min}, d_{max}]$, and employ the occlusion mask M^d to restrict the spatial regions affected by each rain layer. By incorporating these constraints, our method yields a spatially varying rain distribution, where rain streaks appear sparser in the foreground and accumulate more densely in the background. This naturally mirrors complex occlusion characteristics, effectively bridging the domain gap for real-world scenarios.

4 Consistency-Guided Recovery Network

4.1 Motivation

To achieve effective LF deraining, we explore *1) how to detect the region overlapped with rains* and *2) how to fully aggregate the information from non-degraded areas between views for recovery*.

Since views are taken from different angulars of same scene, they are expected to satisfy the disparity constraint:

$$I_{\mathbf{u_j}}(\mathbf{x_j}) = I_{\mathbf{u_i}}(\mathbf{x_i}), \quad \mathbf{x_j} = \mathbf{x_i} + D_{\mathbf{u_i} \to \mathbf{u_j}}(\mathbf{x_i}), \tag{3}$$

where $D_{\mathbf{u_i} \to \mathbf{u_j}}(\mathbf{x_i})$ is derived as:

$$D_{\mathbf{u_i} \to \mathbf{u_j}}(\mathbf{x_i}) = (\mathbf{u_j} - \mathbf{u_i}) * \frac{b * f}{Z(\mathbf{x_i})} = (\mathbf{u_j} - \mathbf{u_i}) * d(\mathbf{x_i}), \tag{4}$$

where b and f represent the baseline length and focal length. $Z(\mathbf{x_i})$ is the depth of $\mathbf{x_i}$. Theoretically, the imaging of all Lambertian scenes follows the above constraint, including the clean background I^b and single rain layer I^r. In real-world scenes, the rain layer naturally overlaps with the background layer, resulting in I^a. However, unlike other occluding objects, rain streaks demonstrate a discrete spatial distribution pattern and semi-translucent properties. This unique translucency allows partial background visibility while degrading the background information. Suppose that one background point is not occluded by any rain streaks in $I^a_{\mathbf{u_i}}$, its pixel intensity is expressed as:

$$I^a_{\mathbf{u_i}}(\mathbf{x_i}) = I^b_{\mathbf{u_i}}(\mathbf{x_i}). \tag{5}$$

When observed from view $\mathbf{u_j}$, the corresponding point is occluded by the rain streak, whose intensity is aliased with rain artifacts:

$$I^a_{\mathbf{u_j}}(\mathbf{x_j}) = I^b_{\mathbf{u_j}}(\mathbf{x_j}) + \alpha \cdot I^r_{\mathbf{u_j}}(\mathbf{x_j}) \tag{6}$$

Combining with Eq. 5 and Eq. 6, it is mathematically derivable that the disparity constraint in Eq. 3 is broken for this point:

$$I^a_{\mathbf{u_j}}(\mathbf{x_j}) = I^a_{\mathbf{u_i}}(\mathbf{x_i}) + \alpha \cdot I^r_{\mathbf{u_j}}(\mathbf{x_j}) \neq I^a_{\mathbf{u_i}}(\mathbf{x_i}) \tag{7}$$

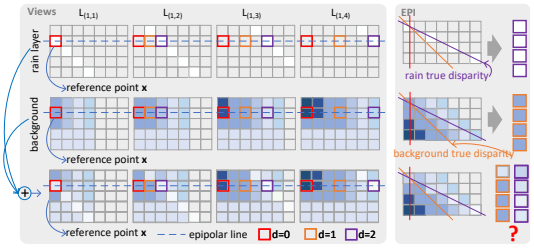

Figure 2: Illustration of epipolar consistency in the rain layer (top), background (middle), and superimposed image (bottom).

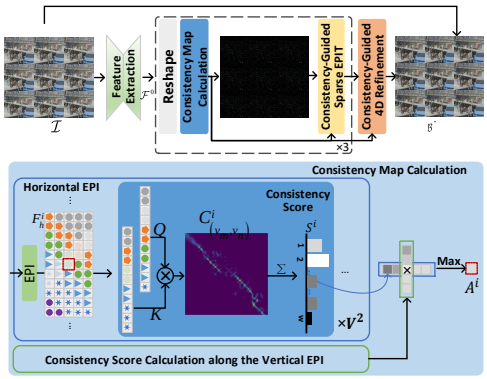

Figure 3: Overview of the proposed CGRNet and illustration of the consistency map calculation.

An example is illustrated in Fig. 2. When considered as independent layers, the rain (purple) and background (orange) pixels at $\mathbf{x}$ shift across views from $L_{(1,1)}$ to $L_{(1,4)}$ according to their respective disparities. These spatial correspondences form continuous lines in the EPIs, with slopes determined by $d^r(\mathbf{x}) = 2$ and $d^b(\mathbf{x}) = 1$. After overlapping, $\mathbf{x}$ is aliased by a rain streak in view $L_{(1,1)}$. Due to the larger disparity of the rain layer ($d^r > d^b$), the rain streak occludes different background areas in the other views ($L_{(1,2)}$, $L_{(1,3)}$, and $L_{(1,4)}$). Consequently, the corrupted pixel at $L_{(1,1)}$ lacks consistent cross-view matches, causing discontinuities in the EPI lines. In contrast, the background information may clearly visible in views $L_{(1,2)}$, $L_{(1,3)}$, and $L_{(1,4)}$, providing clean redundancy that can be leveraged to restore the degraded pixel at $L_{(1,1)}$.

Therefore, 1) inspired by the inconsistency introduced by rain, we propose to measure epipolar consistency to locate the degraded regions. 2) Since the degradation level for one point is different among views, we propose to explore the non-degraded or light degraded information for recovery.

4.2 Network Architecture

As illustrated in Fig. 3, the input rainy LF $\mathcal{I}$ is first processed by a U-Net backbone to extract multi-scale representations of rain degradation, denoted as $\mathcal{F}^0$. Then, we introduce the *Consistency Map Calculation* module that encode the degree of degradation via the epipolar consistency. Guided by the consistency map, our *Consistency-Guided Sparse EPI Transformer* and *Consistency-Guided 4D Refinement* modules selectively aggregate clean features across the 4D LF space for robust deraining.

4.2.1 Consistency Map Calculation

We propose to encode the degree of degradation by measuring epipolar consistency along horizontal and vertical epipolar lines as shown in Fig. 3. First, we use the EPI parallax attention [10] to compute the correlation between pixels along the epipolar lines. Taking the horizontal EPI as an example. For the given feature $\mathcal{F}^i$, where $i = 1, 2, 3$ denotes the iteration number, we reshape and split it into UH horizontal EPI features $F_h^i \in \mathbb{R}^{V \times W \times C}$. Then, each horizontal EPI feature is fed into 2D convolution layers and processed by a self-attention mechanism to calculate pixel correlations. This porcess yields $V \times V$ parallax-attention maps $C^i_{(v_m,v_n)} \in \mathbb{R}^{W \times W}$ to record the pixel-wise similarities between the horizontal epipolar lines of view v_m and view v_n. Subsequently, we apply a disparity constraint to the $C^i_{(v_m,v_n)}$ (as in [10]) and sum the similarities across all candidate matches to derive the consistency score for pixels in view v_m relative to v_n, expressed as $\sum_{w\in[1,W]} C^i_{(v_m,v_n)}(h, w)$. Since similarities in inconsistent areas remain low across all disparities, this aggregated score acts as a probabilistic measure: a lower score indicates a reduced probability of finding consistent information in the target view. Similarly, applying corresponding operations to vertical EPI features $F_v^i \in \mathbb{R}^{U \times H \times C}$ yields U vertical consistency scores for per pixel, encoding the matching probability across vertical views. The horizontal and vertical consistency scores are concatenated, yielding the $\mathcal{S}^i \in \mathbb{R}^{U \times V \times H \times W \times (U+V)}$. However, in addition to rain degradation, other factors like ambiguities, occlusion, or noise also cause low consistency. Compared with rain degradation, these factors usually only cause inconsistency in

Table 2: Quantitative comparison on synthetic and real-world datasets. The best results are bolded.

Method	Synthetic (PSNR↑/SSIM↑/LPIPS↓)			Real (NIQE↓/BRISQUE↓)			Para. (M)	Time (S)
	LFDR-Light	LFDR-Heavy	LFDR-STF	LFRR-Light	LFRR-Heavy	RLMB-Real		
Restormer* [40]	39.60 / 0.9781 / 0.0174	33.46 / 0.9252 / 0.0703	33.28 / 0.9075 / 0.1052	14.51 / 47.56	17.97 / 54.20	20.05 / 40.34	26.12	15.14
DRSformer* [3]	37.64 / 0.9692 / 0.0228	31.99 / 0.9012 / 0.0942	31.20 / 0.8645 / 0.1498	15.72 / 51.81	18.62 / 57.93	20.70 / 31.90	33.65	35.15
NeRD* [4]	39.32 / 0.9765 / 0.0181	33.68 / 0.9283 / 0.0835	33.30 / 0.9077 / 0.1019	14.78 / 44.55	17.66 / 51.15	19.18 / 29.56	22.89	19.57
EPIT [19]	40.58 / 0.9724 / 0.0091	32.94 / 0.9204 / 0.0603	32.98 / 0.8636 / 0.1462	14.46 / 45.84	15.22 / 49.14	18.62 / 37.57	**3.28**	**3.00**
MGPD [31]	38.42 / 0.9653 / 0.0129	31.85 / 0.8903 / 0.0690	31.56 / 0.8389 / 0.1676	14.78 / 47.69	15.72 / 52.45	15.77 / 38.54	14.91	25.86
MDerainNet [32]	39.40 / 0.9712 / 0.0121	34.29 / 0.9295 / 0.0451	34.03 / 0.9020 / 0.1293	14.21 / 41.44	14.85 / 44.62	15.12 / 37.59	315.91	19.04
CGRNet (Ours)	**41.33 / 0.9849 / 0.0062**	**34.79 / 0.9362 / 0.0438**	**34.50 / 0.9107 / 0.0943**	**12.45 / 38.70**	**13.77 / 42.66**	**14.45 / 28.75**	3.57	3.31

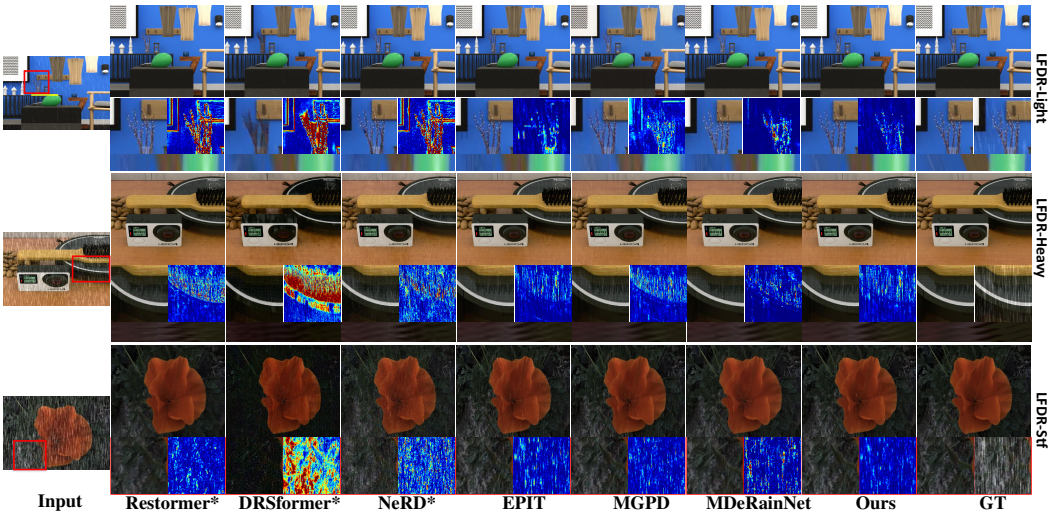

Figure 4: Visual comparison on the *LFDR-Light*, *LFDR-Heavy* and *LFDR-STF*. The central view and EPIs of result are shown. Red indicates a large discrepancy between output and GT in error map.

one or part view. For the background occluded by rain streaks, it is challenging to find consistent matching information across all views. Therefore, the consistency maps $\mathcal{A}^i$ are integrated by getting the maximum similarity across views:

$$\mathcal{A}^i = \mathcal{S}^i(:,:,:,:,j^*), \tag{8}$$

where j^* represents the view position with the highest consistency score:

$$j^* = \arg\max_j(\mathcal{S}^i(:,:,:,:,j))). \tag{9}$$

4.2.2 Consistency-Guided Recovery Module

Due to rain streak transparency, degraded regions often appear excessively bright, and utilizing these areas exacerbates rain artifacts. Therefore, it is crucial to selectively utilize non-degraded or lightly degraded cues for restoration. Based on the consistency map, we propose the *Consistency-Guided Sparse EPI Transformer* and *Consistency-Guided 4D Refinement* modules.

Operating on horizontal or vertical EPIs, the *Consistency-Guided Sparse EPI Transformer* filters out unnecessary tokens according to the consistency maps for sparse reliable recovery. For each epipolar line, we select the top-K pixels with the highest consistency scores to form a clean feature set after feature embedding. Then, the original features are projected to generate the Query, while the selected sparse clean features are projected to generate the Key and Value. By calculating the self-attention between them, the model effectively aggregates redundant clean cues for detail reconstruction, yielding the initial output $\hat{\mathcal{B}}_{init}$.

However, severely degraded areas are difficult to restore using only the clean cues in the EPIs. To address this, the *Consistency-Guided 4D Refinement* module samples similar clean cues across the 4D LF space. Specifically, to effectively focus on non-degraded regions, we concatenate the consistency

Table 3: Human perceptual evaluation on real-world scenes. Preference rates (%) are reported.

Method	C1	C2	C3	C4	C5
Restormer*	2.7	0.9	1.5	1.0	–
DRSformer*	1.8	0.6	1.0	0.8	–
NeRD*	2.2	0.8	1.3	1.1	–
EPIT	1.2	2.3	3.3	3.8	1.6
MGPD	0.9	2.0	3.0	3.4	1.4
MDeRainNet	2.2	5.9	7.9	9.4	4.5
CGRNet (Ours)	**89.0**	**87.5**	**82.0**	**80.5**	**92.5**

Table 4: Consistency map and depth estimation evaluation.

Task	**Method**	**LFDR-Light**	**LFDR-Heavy**
Consistency Map (AP↑)	JORDER	0.554	0.512
	S2VD	0.487	0.501
	Ours	**0.698**	**0.735**
Depth Estimation	EPIT	2.73/16.28	2.79/31.94
	MGPD	1.75/16.08	2.56/31.21
	MDeRainNet	1.00/11.65	2.49/26.41
	Ours	**0.81/6.85**	**2.12/20.33**

maps with the input features to predict 4D offsets. These learned offsets are then used to guide the spatial-angular deformation for each pixel. Finally, the enhanced features are aggregated to produce the refined output $\hat{\mathcal{B}}$ (further details are provided in the Appendix B).

4.3 Implementation Details

We train CGRNet by minimizing a hybrid loss: $\ell_{total} = \lambda_1 \ell_{rec_1} + \lambda_2 \ell_{rec_2} + \lambda_3 \ell_{PAM}$. ℓ_{rec_1} and ℓ_{rec_2} are ℓ_1 reconstruction losses between the ground truth (GT) clean image $\mathcal{B}$ with the intermediate predictions $\hat{\mathcal{B}}_{init}$ and refined predictions $\hat{\mathcal{B}}$, respectively. ℓ_{PAM} is the PAM loss from [26] to ensure the accuracy of $\mathcal{S}$. Hyperparameters λ_1, λ_2, and λ_3 are empirically set to 0.4, 0.6, and 0.2.

Our model is implemented by the PyTorch framework and trained on an NVIDIA RTX A4000 GPU. We employ the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. During training, the central 5×5 views are extracted from each LF, and randomly cropped into 64×64 patches. The learning rate is set to $1e-4$ and decreased by a factor of 0.5 every 500 epochs. The batch size is set to 1. Data augmentation techniques such as random cropping, vertical and horizontal flipping are applied.

5 Experiments

5.1 Dataset

We evaluate our method on our proposed synthetic dataset *LFDR* and the real-world dataset *LFRR*, both of which will be publicly released to support future research in LF deraining. Additionally, we evaluate our method on the RLMB-Real [31] to validate its generalization to real-world scenes.
Synthetic dataset LFDR: LFDR is a dataset generated using the model described in Sec. A, comprising 240 images for training and 60 for testing. The training images are synthesized using 240 scenes from the HCI [13, 29], DLFD [22], and UrbanLF-Syn-Big [21] dataset as $\mathcal{B}$ in Eq. 1. These datasets provide available depth maps to calculate the occlusion masks. In addition, we select 19 scenes from the aforementioned datasets to construct the test set, which is synthesized under both light and heavy rain conditions (by varying rain layer numbers) to form the *LFDR-Light* and *LFDR-Heavy*. The *LFDR-Light* is also used as the validation set. To evaluate the generalization performance, an additional 22 scenes are synthesized based on the real-world Stanford dataset [24], with depth maps estimated using the method in [10]. This subset is referred to as *LFDR-STF*.
Real-world dataset LFRR: To evaluate performance on real-world scenes, we capture 30 LF images under light and heavy rain, forming two subsets of 15 samples each: *LFRR-Light* and *LFRR-Heavy*.

5.2 Comparison with SOTA Methods

We compare our method with four single-frame-based deraining methods (i.e. Restormer [40], DRSformer [3], NeRD [4], ESDNet [23]), which are carefully fine-tuned on our *LFDR* (denoted as *) to ensure a fair comparison. During training, each view of the LF is treated as an independent sample, resulting in a total of 19,440 $(9 \times 9 \times 240)$ image pairs. The model configurations follow the original papers. We also adjust the commonly used LF image processing frameworks SAS [36], EPIT [19] for the LF deraining. The LF deraining methods MGPD [31] and MDeRainNet [32] are also compared.

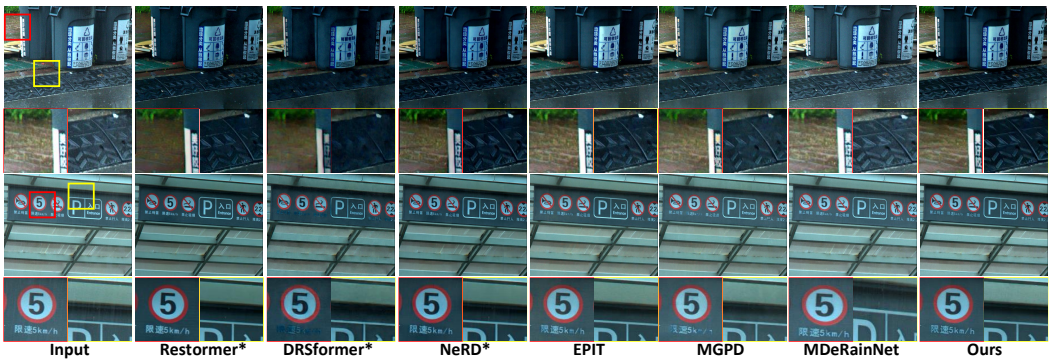

Figure 5: Visual comparison on the *LFRR-Light*. Details are magnified for clearer comparison.

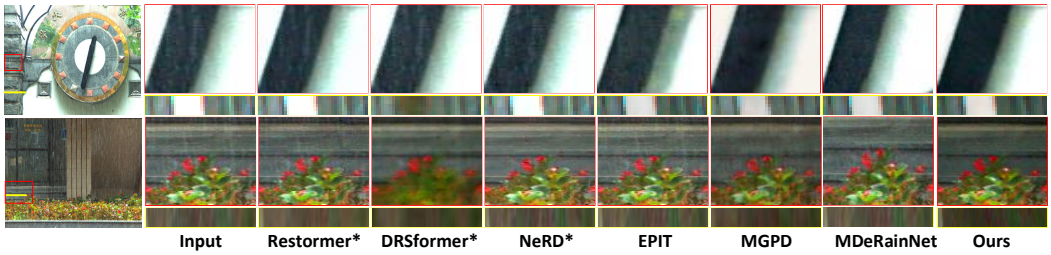

Figure 6: Visual comparison on *LFRR-Heavy*. Central view and EPIs of the recovered LFs are shown.

All LF-based methods are trained using *LFDR* dataset. Both reference-based metrics (PSNR, SSIM and LPIPS) and no-reference metrics (NIQE and BRISQUE) are adopted for performance evaluation.

Comparison on Synthetic Scenes. Table 2 presents a quantitative comparison between our method and existing deraining methods on *LFDR*. Compared with single-frame-based methods, LF-based methods achieve higher performance since the abundant information across views in LFs is beneficial to recover the degraded details. Among LF-based methods, our CGRNet achieves superior metric results on datasets in both heavy and light rains with fewer parameters. Specifically, our CGRNet outperforms the latest LF deraining model, MDeRainNet, by average margins of 0.96 dB in PSNR, while requiring only 3.57M parameters. Compared to the general LF processing method EPIT, it achieves significant PSNR improvements of 0.75 dB, 1.85 dB, and 1.52 dB across the three datasets, respectively. The significant gains on *LFDR-Heavy* demonstrate our method's robustness to heavy degradation. Furthermore, achieving the highest PSNR/SSIM on *LFDR-STF*, where backgrounds diverge significantly from the training set, highlights the superior generalization of CGRNet. Regarding the perceptual metric LPIPS, our method achieves the lowest score on all dataset, indicating that our recovered images are not only pixel-wise accurate but also perceptually more natural. Fig. 4 displays the visual comparison. While single-frame methods successfully remove rain streaks, they often lead to artifacts and texture distortion in occluded regions. As shown in the middle row of Fig. 4, although these methods remove rain streaks, they severely compromise the texture of the brush reflection. In contrast, LF-based methods better preserve intricate textures by leveraging inter-view redundancy. Their error maps are predominantly blue, indicating lower residual errors. Among them, our CGRNet produces finer details with minimal errors and superior spatial consistency, closely matching the GT.

Real-word Scene. To verify the generalization on real-world scenes, we further present quantitative evaluations on *LFRR* and *RLMB-Real* [31] in Table 2. The models trained on *LFDR* are directly test on these scenes. Our CGRNet consistently achieves the best non-reference metrics across both datasets. Notably, under heavy rain, our method significantly outperforms MDeRainNet, reducing NIQE and BRISQUE by 1.08 and 1.96, respectively. Visual comparison are present in Fig. 5 and Fig. 6. Other approaches either struggle to remove complex degradations or significantly compromise image details. SAS and EPIT leave distinct residual rain streaks, whereas MGPD and MDeRainNet suffer from over-smoothing with significant blur and unfilled areas. Compared with them, our approach restores intricate textures with higher fidelity by modeling degradation relationships and leveraging clean information.

Table 5: Results of module variants. T and B denote top-k and bottom-k token selection, respectively.

No.	Consistency Map	Token Selection		4D Refinement	PSNR/SSIM
		Strategy	Ratio		
(1)	×	–	100%	×	32.94/0.9204
(2)	*min*	T	50%	×	32.72/0.9173
(3)	*mean*	T	50%	×	33.04/0.9214
(4)	*median*	T	50%	×	33.25/0.9223
(5)	*soft weight*	T	50%	×	33.52/0.9231
(6)	*max*	B	50%	×	32.85/0.9189
(7)	*max*	T	50%	×	33.87/0.9248
(8)	*max*	T	50%	w/o guide	34.57/0.9296
(9)	*max*	T	50%	w/ guide	34.79/0.9362
(10)	*max*	T	60%	w/ guide	**34.80/0.9362**
(11)	*max*	T	70%	w/ guide	34.56/0.9292

For instance, in the bottom row of Fig. 6, unlike other methods that degrade the background, our model removes occluding rain streaks while maintaining the sharp textures of the flower. To further assess the perceptual quality of our restoration results, we conducted a randomized and anonymized human perceptual study involving 220 participants and 30 real-world LF rainy scenes. The evaluation considered five criteria: rain removal effectiveness (C1), detail and structure preservation (C2), color and illumination fidelity (C3), artifact suppression (C4), and cross-view consistency (C5). The presentation order was randomized to mitigate positional bias. As shown in Table 3, our method achieved the highest preference rates across all five criteria, with 89.0%, 87.5%, 82.0%, 80.5%, and 92.5%, respectively. More results are provided in the Appendix C.2.

Consistency Map Evaluation. To validate the accuracy of our consistency map, we compared it against JORDER [34] and S2VD [38] on the *LFDR-Light* and *LFDR-Heavy* datasets. Since high consistency indicates a clean pixel, we invert our consistency map to obtain the rain probability. Subsequently, we calculated the Average Precision (AP) for the predictions of all methods against the GT rain residuals ($|\mathcal{I} - \mathcal{B}|$). As shown in Table 4 , our consistency map outperforms existing methods. On the *LFDR-Heavy* characterized by severe rain degradation, it achieves a high localization accuracy with an AP of 0.735, demonstrating its robustness in detecting dense rain streaks.

Epipolar Consistency Comparison. As shown in Figs. 4 and 6, our approach preserves the structural integrity of EPIs, yielding continuous slopes in textured regions. To further demonstrate practical applicability, we input the results into a depth estimation network [28]. As shown in Table 4, our method achieves lowest MSE and BadPixel (lower is better) on *LFDR* datasets, reducing MSE by 0.19/0.37 and BadPixel by 4.80%/6.08% under Light/Heavy conditions compared to MDeRainNet. Fig. 7 illustrates the visual results on the *LFDR-STF*. Benefiting from the selective use of non-degraded regions for recovery, the depth maps generated from our CGRNet outputs are cleaner.

5.3 Ablation Study

We conduct the ablation studies on *LFDR-Heavy* to demonstrate the effectiveness of different components in our model. The parameter of the variant models are closed to full model by adjusting the number of feature channels. The result are peresent in Table 5. First, we remove the consistency maps and the 4D refinement in Model-1, using only the EPIT for recovery, which performs the worst. Compared with Model-7, alternative strategy including the minimum (Model-2), mean (Model-3), and median (Model-4) may incorrectly classify non-degraded regions as degraded regions by overemphasizing the low scores calculated from degraded views, resulting in PSNR drops of 1.15 dB, 0.83 dB, and 0.62 dB, respectively. While Model-5 (soft weight) emphasizes larger values via softmax, the assigned weights from low-consistency (degraded) views inevitably dilute the contribution of the few high-consistency (clean) views. Consequently, it still underperforms Model-7. Compared to Model-7, Model-6 selects the bottom 50% of low-consistency tokens for recovery. Since these tokens correspond to degraded regions, Model-6 incorporates more corrupted information, resulting in performance even worse than Model-1.

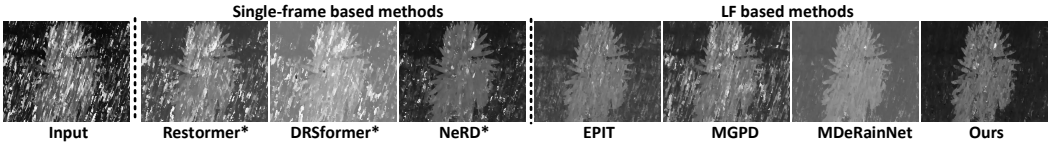

Figure 7: Visual comparison of depth estimation results of *LFDR-STF*.

Table 6: Quantitative comparison under different angular resolutions (PSNR↑/SSIM↑/LPIPS↓).

Method	3×3	5×5	7×7
EPIT	32.55/0.9158/0.0658	32.94/0.9204/0.0603	33.11/0.9226/0.0584
CGRNet (Ours)	**34.31/0.9308/0.0497**	**34.79/0.9362/0.0438**	**35.00/0.9384/0.0418**

Table 7: Evaluation of models trained on RLMB-SYN and LFDR using NIQE↓/BRISQUE↓.

Training Set	LFRR-Light		LFRR-Heavy		RLMB-Real	
	MGPD	CGRNet	MGPD	CGRNet	MGPD	CGRNet
RLMB-SYN	17.50/51.59	15.85/37.78	17.04/54.58	15.30/45.42	19.29/44.27	18.64/37.78
LFDR	14.78/47.69	12.45/38.70	15.72/52.45	13.77/42.66	15.77/38.54	14.45/28.75

In Model-8, we add a 4D refinement module without guidance from the consistency maps, leading to a PSNR improvement of nearly 1 dB. This indicates that our refinement module is able to further enhance details in degraded regions. Compared to Model-8, Model-9 achieves even better performance by leveraging consistency maps to guide offset generation, highlighting the effectiveness of consistency-guided offset estimation.

We also design Model-9, Model-10, and Model-11 to investigate the influence of the parameter K in Eq. 12. Model-9 achieves the best performance when K is set to 50%. When K is small, the utilization of clean cues is insufficient, whereas when K is large, some degraded region information is incorrectly introduced. In Table 6, we additionally evaluate 3×3, 5×5, and 7×7 angular resolutions on *LFDR-Heavy*. Increasing the number of views consistently improves performance by providing richer angular information.

5.4 Dataset Validation

To demonstrate the superiority of dataset constructed using our consistency-constrained LF rain model, we train our model and MGPD on both *LFDR* and *RLMB-SYN* [31] datasets and test on the real-world dataset. Following [31], we use 300 synthetic images for training in the *RLMB-SYN* dataset. As shown in Table 7, both MGPD and CGRNet trained on our *LFDR* dataset yield better generalization. Specifically, on the *RLMB-Real* test set, LFDR-trained models reduces the NIQE/BRISQUE scores of MGPD and our CGRNet to 15.77/38.54 and 14.45/28.75, respectively, which are notably better than the 19.29/44.27 and 18.64/37.78 achieved by the RLMB-SYN trained models. Similar consistent improvements are also observed on the *LFRR* datasets.

6 Conclusion

In this paper, we explore relation between the consistency and degradation. Based on the consistency, we construct a consistency-constrained Light Field (LF) rain model and presented a transform-based framework for LF deraining. We introduce the consistency map to model scene degradation. Based on the consistency maps, we design a sparse sampling transformer and a 4D refinement module to aggregate reliably clean information across views. In addition, we synthesize a more realistic rainy LF dataset according to the proposed rain model. Quantitative and qualitative comparisons show that our model achieves state-of-the-art performance compared to other methods, demonstrating the effectiveness of our dataset.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62372032).

References

[1] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12294–12305, 2021. doi: 10.1109/CVPR46437.2021.01212.

[2] S. Chen, T. Ye, J. Bai, E. Chen, J. Shi, and L. Zhu. Sparse sampling transformer with uncertainty-driven ranking for unified removal of raindrops and rain streaks. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13060–13071, 2023. doi: 10.1109/ICCV51070.2023.01205.

[3] X. Chen, H. Li, M. Li, and J. Pan. Learning a sparse transformer network for effective image deraining. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5896–5905, 2023. doi: 10.1109/CVPR52729.2023.00571.

[4] X. Chen, J. Pan, and J. Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25627–25636, 2024. doi: 10.1109/CVPR52733.2024.02421.

[5] X. Chen, J. Pan, J. Dong, and J. Tang. Towards unified deep image deraining: A survey and a new benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5414–5433, 2025. doi: 10.1109/TPAMI.2025.3556133.

[6] B. O. Community. Blender - a 3d modelling and rendering package. `https://www.blender.org/`, 2018. Accessed: Jan. 12, 2022.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. doi: 10.1109/ICCV.2017.89.

[8] Y. Ding, M. Li, T. Yan, F. Zhang, Y. Liu, and R. W. H. Lau. Rain streak removal from light field images. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2):467–482, 2022. doi: 10.1109/TCSVT.2021.3063853.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[10] C. Gao, S. Zhang, and Y. Lin. Epipolar consistent attention aggregation network for unsupervised light field disparity estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6488–6497, October 2025.

[11] K. Garg and S. K. Nayar. Vision and rain. *Int. J. Comput. Vision*, 75(1):3–27, Oct. 2007. ISSN 0920-5691. doi: 10.1007/s11263-006-0028-6. URL `https://doi.org/10.1007/s11263-006-0028-6`.

[12] S. S. Halder, J.-F. Lalonde, and R. de Charette. Physics-based rendering for improving robustness to rain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10203–10212, 2019.

[13] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13*, pages 19–34. Springer, 2017.

[14] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8022–8031, June 2019.

[15] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang. Multi-scale progressive fusion network for single image deraining. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8343–8352, 2020. doi: 10.1109/CVPR42600.2020.00837.

[16] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996.

[17] P. Li, J. Jin, G. Jin, L. Fan, X. Gao, T. Song, and X. Chen. Deep scale-space mining network for single image deraining. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4275–4284, 2022. doi: 10.1109/CVPRW56347.2022.00473.

[18] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *European Conference on Computer Vision*, pages 262–277. Springer, 2018.

[19] Z. Liang, Y. Wang, L. Wang, J. Yang, S. Zhou, and Y. Guo. Learning non-local spatial-angular correlation for light field image super-resolution. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12342–12352, 2023. doi: 10.1109/ICCV51070.2023.01137.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[21] H. Sheng, R. Cong, D. Yang, R. Chen, S. Wang, and Z. Cui. Urbanlf: A comprehensive light field dataset for semantic segmentation of urban scenes. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7880–7893, 2022. doi: 10.1109/TCSVT.2022.3187664.

[22] J. Shi, X. Jiang, and C. Guillemot. A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Transactions on Image Processing*, 28(12):5867–5880, 2019. doi: 10.1109/TIP.2019.2923323.

[23] T. Song, G. Jin, P. Li, K. Jiang, X. Chen, and J. Jin. Learning a spiking neural network for efficient image deraining. In K. Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 1254–1262. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/139. URL `https://doi.org/10.24963/ijcai.2024/139`. Main Track.

[24] A. Sunder Raj, M. Lowney, R. Shah, and G. Wetzstein. The stanford lytro light field archive. `http://lightfields.stanford.edu/LF2016.html`, 2016. Accessed: Aug. 22, 2020.

[25] H. Wang, Y. Wu, M. Li, Q. Zhao, and D. Meng. Survey on rain removal from videos or a single image. *Science China Information Sciences*, 65(1), Dec. 2021. ISSN 1869-1919. doi: 10.1007/s11432-020-3225-9. URL `http://dx.doi.org/10.1007/s11432-020-3225-9`.

[26] L. Wang, Y. Guo, Y. Wang, Z. Liang, Z. Lin, J. Yang, and W. An. Parallax attention for unsupervised stereo correspondence learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4): 2108–2125, 2022. doi: 10.1109/TPAMI.2020.3026899.

[27] Y. Wang, Y. Song, C. Ma, and B. Zeng. Rethinking image deraining via rain streaks and vapors. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 367–382, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58520-4.

[28] Y. Wang, L. Wang, Z. Liang, J. Yang, W. An, and Y. Guo. Occlusion-aware cost constructor for light field depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19809–19818, June 2022.

[29] S. Wanner, S. Meister, and B. Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In *VMV*, volume 13, pages 225–226, 2013.

[30] J. Xiao, X. Fu, A. Liu, F. Wu, and Z.-J. Zha. Image de-raining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12978–12995, 2023. doi: 10.1109/TPAMI.2022.3183612.

[31] T. Yan, M. Li, B. Li, Y. Yang, and R. W. H. Lau. Rain removal from light field images with 4d convolution and multi-scale gaussian process, 2023.

[32] T. Yan, W. Huang, W. He, C. Wang, C. Wei, Y. Lu, X. Zhu, Y. Wang, and R. W. H. Lau. Mderainnet: An efficient macro-pixel image rain removal network. *ACM Trans. Multimedia Comput. Commun. Appl.*, 22 (2), Feb. 2026. ISSN 1551-6857. doi: 10.1145/3775059. URL `https://doi.org/10.1145/3775059`.

[33] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017. doi: 10.1109/CVPR.2017.183.

[34] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (6):1377–1393, 2020. doi: 10.1109/TPAMI.2019.2895793.

[35] W. Yang, R. T. Tan, J. Feng, S. Wang, B. Cheng, and J. Liu. Recurrent multi-frame deraining: Combining physics guidance and adversarial learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8569–8586, 2022.

[36] H. W. F. Yeung, J. Hou, X. Chen, J. Chen, Z. Chen, and Y. Y. Chung. Light field spatial super-resolution using deep efficient spatial-angular separable convolution. *IEEE Transactions on Image Processing*, 28(5): 2319–2330, 2019. doi: 10.1109/TIP.2018.2885236.

[37] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. doi: 10.1109/CVPR.2018.00212.

[38] Z. Yue, J. Xie, Q. Zhao, and D. Meng. Semi-supervised video deraining with dynamical rain generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 642–652, 2021. doi: 10.1109/CVPR46437.2021.00070.

[39] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Multi-stage progressive image restoration. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14816–14826, 2021. doi: 10.1109/CVPR46437.2021.01458.

[40] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. doi: 10.1109/CVPR52688.2022.00564.

[41] H. Zhang and V. M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 695–704, 2018. doi: 10.1109/CVPR.2018.00079.

[42] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2023.

[43] K. Zhang, D. Li, W. Luo, W. Ren, and W. Liu. Enhanced spatio-temporal interaction learning for video deraining: Faster and better. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 1287–1293, 2023. doi: 10.1109/TPAMI.2022.3148707.

A Synthesis Pipeline

Building uponour consistency-constrained LF rain model, we develop a consistency-constrained LF rain synthesis pipeline (Algorithm. 1) that generates plausible rainy LF images, which takes the clean LF image $\mathcal{B}$ and its central-view disparity map $D_{\mathbf{u_c}}$ as input to generate realistic rainy LF image $\mathcal{I}$.

For a given scene, we first initialize the synthetic LF image $\mathcal{I}^0$ as the clean input $\mathcal{B}$ and extract the $d_{min}\&d_{max}$ from $D_{\mathbf{u_c}}$ to define the range of depth over which rain layers will be applied. This range is then discretized into uniform intervals using a step size Δd, determining both the number and granularity of rain layers. Subsequently, we perform *Rain Layer Generation* and *Binary Mask Generation* at each predefined depth to enforce geometric consistency withi Finally, *Rendering and Layer Fusion* are applied to integrate the rain layers from far to near by superimposing the corresponding masks at each depth, ensuring the physical plausibility of accumulation and spatial coherence.

Rain Layer Generation: In this step, we first generate the rain layers for the central view and then shift them according to depth to ensure the rain layers consistency. For each depth layer $d_i = d_{min} + i \cdot \Delta d$, the appearance of the rain layer is explicitly controlled by four key parameters: intensity ϕ_i, density ρ_i, angle θ_i, and length l_i. Rather than treating the parameters of each layer independently, we linearly update them across layers using slight step sizes to reflect the physical reality that rain properties change with depth [11, 33]. To further enhance realism, we introduce noise into each rain layer to simulate environmental factors such as wind perturbations and imaging artifacts. Finally, these rain layers are shifted to all views $\mathbf{u}$ to maintain consistency and ensure coherent rain geometry across the LF, as described in Eq. 2. As shown in Fig. A, we evaluate the influence of individual parameters by analyzing the pixel-wise error of generated dataset. To modulate the different visual degradation (ϕ, ρ, l) of the scene, we maintain constant step sizes across layers while strictly varying the settings of the 1-th rain layer. Specifically, higher rain intensity (ϕ) and density (ρ) significantly increase the rain coverage, which severely exacerbates the degradation in the affected regions (Fig. A(a, b)). Conversely, extending the streak length (l) mitigates the degradation, slightly expanding the high-fidelity areas (Fig. A(c)). Beyond these 2D attributes, we investigate the impact of the Δd by setting different numbers of simulated rain layers, formulated as $(d_{max} - d_{min})/\Delta d$. As expected, an increase in the layer count causes the high-fidelity regions to shrink rapidly (Fig. A(d)).

Binary Mask Generation: As mentioned above, the rain layers overlap only with the background or with other rain layers at deeper depths. To control the spatial regions affected by each rain layer and ensure correct occlusion relationships, we calculate a binary mask $M_{\mathbf{u_c}}^{d_i}$ for the central view based on the disparity map $D_{\mathbf{u_c}}$, as defined by:

$$M_{\mathbf{u_c}}^{d_i}(x, y) = \begin{cases} 1 & D_{\mathbf{u_c}}(x, y) \leq d_i, \\ 0 & D_{\mathbf{u_c}}(x, y) > d_i. \end{cases} \tag{10}$$

This mask indicates which pixels are affected by the rain layer at depth corresponding to d_i. To ensure geometric consistency across views, we shift $M_{\mathbf{u_c}}^{d_i}$ according to the disparity d_i to obtain $M_{\mathbf{u}}^{d_i}$ for each view. This step guarantees that each rain layer is properly aligned with the scene geometry and maintains consistent spatial relationships across all views.

Rendering and Layer Fusion: After generating the rain layers$\{R_{\mathbf{u}}^{d_i}\}_{d_i}$ and corresponding masks $\{M_{\mathbf{u}}^{d_i}\}_{d_i}$ for each view, we iteratively blend each rain layer into the LF images. This process is performed from the farthest depth ($d_{\min}$) to the nearest ($d_{\max}$), progressively accumulating rain while

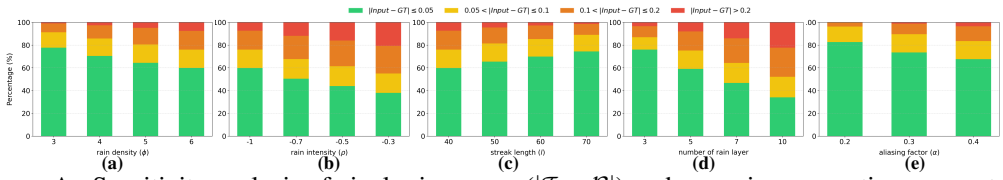

Figure A: Sensitivity analysis of pixel-wise errors ($|\mathcal{I} - \mathcal{B}|$) under varying generation parameters: (a) rain density (ϕ), (b) rain intensity (ρ), (c) streak length (l), (d) number of rain layer ($(d_{max} - d_{min})/\Delta d$), and (e) aliasing factor (α). Stacked bars denote the percentage of pixels in specific error intervals (e.g., green indicates high-fidelity regions, error ≤ 0.05).

Algorithm 1 Consistency-Constrained rain LF generation

Input: LF images $\mathcal{B}$, disparity map of central view $D_{\mathbf{u_c}}$, step size Δd, initial parameters $(\phi, \rho, l, \theta, \alpha)$

$\mathcal{I}^0 = \mathcal{B}$; $i = 1$; $d_{min} = \lceil \min(D_{\mathbf{u_c}}) \rceil$; $d_{max} = \lceil \max(D_{\mathbf{u_c}}) \rceil$;
while $d_i \leq d_{max}$ **do**
  $d_i = d_{min} + i \times \Delta d$;
  $\{\phi_i, \rho_i, l_i\} \leftarrow \{\phi, \rho, l\}$ **if** $i = 1$ **else** $\{\phi_{i-1}, \rho_{i-1}, l_{i-1}\}$
  Compute $M_{\mathbf{u_c}}^{d_i}$ according to Eq. 10
  $R_{\mathbf{u_c}}^{d_i} \leftarrow \{\phi_i, \rho_i, \theta, l_i\}$
  for $u = 0$ to U **do**
    for $v = 0$ to V **do**
      Compute $M_{\mathbf{u}}^{d_i}, R_{\mathbf{u}}^{d_i}$ according to Eq. 2
      Compute $I_{\mathbf{u}}^{i}$ according to Eq. 11
    end for
  end for
  $i = i + 1$
end while
return $\mathcal{I}$

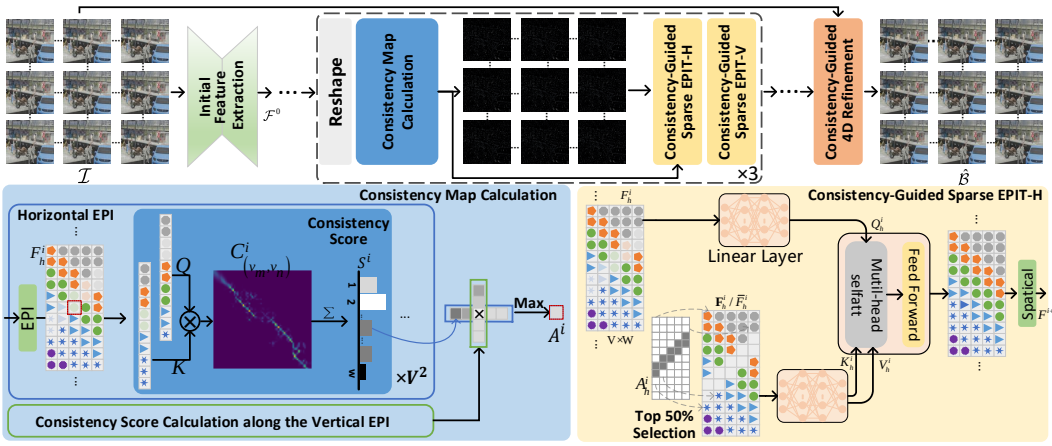

Figure B: An overview of our LF deraining network. After multi-scale feature extraction, consistency maps are constructed to encode degradation levels across different regions. Guided by the consistency maps, the consistency-guided sparse EPI transformer extracts reliable clean information for background recovery. Finally, the deraining result is refined by the consistency-guided 4D refinement module.

preserving occlusion relationships. To simulate atmospheric effects, we also incorporate α to regulate background visibility, following the formulation in [12]:

$$I_{\mathbf{u}}^{i} = I_{\mathbf{u}}^{i-1} + \alpha \cdot R_{\mathbf{u}}^{d_i} \cdot M_{\mathbf{u}}^{d_i}, \tag{11}$$

As shown in Fig. A(e), a higher α reduces streak transparency, resulting in a decrease of the high-fidelity region. After processing all layers, we obtain the $\mathcal{I}$, which exhibits both spatial and angular consistency with a realistic rain appearance.

B Network Architecture

The overall network structure is shown in Fig. B. Our network accepts a rainy LF image $\mathcal{I}$ as input and predicts the clean background $\hat{\mathcal{B}}$ obscured by rain streaks. First, we employ an Unet [20] to excavate the multi-scale representation of rain degradation, denoted as $\mathcal{F}^0$. Next, the module *Consistency Map Calculation* and *Consistency-Guided Sparse EPI Transformer* are alternately employed three times for epipolar consistent clean feature aggregation. Finally, the *Consistency-Guided 4D Refinement* is employed to refine details by sampling relevant spatial information within the 4D LF space.

B.1 Consistency-Guided Sparse EPI Transformer

To effectively aggregate redundant information from clean views, we design the *Consistency-Guided Sparse EPI Transformer* to iterative processing on horizontal EPI and vertical EPI. This module operates in two key steps: (i) *Consistency-guided token selection*, which filters out unreliable tokens from degraded areas based on the consistency maps, and (ii) *Sparse reliable recovery*, which leverages the clean, complementary information along epipolar lines for restoration.

As illustrated in Fig. B , we use the horizontal EPI process as an example. The input features $\mathcal{F}^i$ and consistency maps $\mathcal{A}^i$ are first reshaped into the horizontal EPI pattern $F_h^i \in \mathbb{R}^{UH \times VW \times C}$ and $A_h^i \in \mathbb{R}^{UH \times VW}$, respectively. Since including severely degraded areas in the attention mechanism interferes with restoration, we utilize A_h^i as a reliability selector and perform a consistency-guided token selection to distinguish between degraded and clean regions. Specifically, for each epipolar line, we select the top-K pixels with the highest consistency scores to form the clean feature set $\overline{\mathbf{F}}_h^i$:

$$\overline{\mathbf{F}}_h^i = \{F_h^i(b,p,:) \mid A_h^i(b,p) \geq \tau_b, \quad p \in [1, VW]\}_{b=1}^{UH}, \tag{12}$$

where p indexes the flattened spatial position (v, w), and τ_b denotes the threshold value corresponding to the K-th largest consistency score in the b-th epipolar line. Then, we adopt a sparse reliable recovery strategy to restore the entire scene by exclusively referencing reliable information. We project the original feature F_h^i through a linear layer to generate the dense Query $Q_h^i \in \mathbb{R}^{UH \times VW \times C}$. Simultaneously, the selected sparse features $\overline{\mathbf{F}}_h^i$ are concatenated and projected through two separate linear layers to generate the sparse Key K_h^i and Value V_h^i, both with dimensions $\mathbb{R}^{UH \times K \times C}$. The self-attention is then calculated by interacting the Q_h^i with the K_h^i:

$$\text{Attention}(Q_h^i, K_h^i, V_h^i) = \text{Softmax}\left(\frac{Q_h^i (K_h^i)^\top}{\sqrt{d_k}}\right) V_h^i. \tag{13}$$

In this approach, the model implicitly explores the epipolar consistency among the reliable background tokens in V_h^i, effectively aggregating redundant clean cues for enhanced detail reconstruction.

B.2 Consistency-Guided 4D Refinement

Severely degraded areas affected by rain streaks make it difficult to recover only by clean redundant information in the EPIs. In this case, we further focus on non-degraded information across the entire LF $\hat{\mathcal{B}}_{init}$ and sample similar clean cues within the 4D LF space guided by the consistency map to refine the details.

We first use 2D spatial convolution to encode the $\hat{\mathcal{B}}_{init}$ and obtain the $\mathcal{F}_{init}$. Then, the features of views are stacked and fed to 3D convolutional layers to predict the 4D offset $\Delta p \in \mathbb{R}^{U \times V \times H \times W \times 4\text{N}}$. Here, 4N represents the offset of N sample points across the two angular coordinates $\mathbf{u} = (u, v)$ and two spatial coordinates $\mathbf{x} = (x, y)$. To improve model sampling and more effectively focus on non-degraded regions, we concatenate the consistency maps with features for offset prediction. Then, the learned offsets are used to guide the 4D deformation for each pixel. To enhance sampling effectiveness, we separate 4D deformation into spatial and angular deformation. Specifically, we first perform spatial deformation:

$$F^s(\mathbf{u}, \mathbf{x}) = \sum_{n=1}^{N} w_s(\mathbf{p}_n) \cdot F_{init}(\mathbf{u}, \mathbf{x} + \mathbf{p}_n + \Delta\mathbf{p}_n^s) \tag{14}$$

where $\Delta\mathbf{p}_n^s$ is the predicted spatial offset, w_s represents the weights in the summation, and $\mathbf{p}_n$ represents the n-th value in the sample grid $\mathbf{G} = \{(-1,-1), (-1,0), ..., (1,1)\}$ as in [7]. Next, we perform the deformation along the angular dimension using F^s and predicted angular offset $\Delta\mathbf{p}_n^a$ similarly to Eq. 14. Finally, the enhanced features are fed to a 2D spatial convolution to predict the refined output $\hat{\mathcal{B}}$.

C Experiments

C.1 Comparison on Synthetic Scenes

We further compare our method with two video-based deraining methods (i.e. S2VD [38], ESTINet [43]). To adapt these methods to our task, we sequentially stack the multi-view of LF images

Table A: More result on LFDR. The best results are bolded.

Type	Method	LFDR-Light	LFDR-Heavy	LFDR-STF	Average	Efficiency	
		PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓	PSNR↑/SSIM↑/LPIPS↓	Params (M)	Time (s)
Single-frame based	Restormer* (CVPR 2022)	39.60 / 0.9781 / 0.0174	33.46 / 0.9252 / 0.0703	33.28 / 0.9075 / 0.1052	35.45 / 0.9369 / 0.0643	26.12	15.14
	DRSformer* (CVPR 2023)	37.64 / 0.9692 / 0.0228	31.99 / 0.9012 / 0.0942	31.20 / 0.8645 / 0.1498	33.61 / 0.9116 / 0.0889	33.65	35.15
	NeRD* (CVPR 2024)	39.32 / 0.9765 / 0.0181	33.68 / 0.9283 / 0.0835	33.30 / 0.9077 / 0.1019	35.43 / 0.9375 / 0.0678	22.89	19.57
	ESDNet* (IJCAI 2024)	38.52 / 0.9876 / 0.0167	33.29 / 0.9543 / 0.0686	33.85 / 0.9188 / 0.1256	35.22 / 0.9536 / 0.0703	12.75	2.37
Video based	S2VD* (CVPR 2021)	37.41 / 0.9658 / 0.0432	30.00 / 0.8514 / 0.1726	30.23 / 0.8261 / 0.2160	32.55 / 0.8811 / 0.1439	**0.50**	6.59
	ESTINet* (TPAMI 2023)	37.81 / 0.9701 / 0.0207	33.83 / 0.9128 / 0.0981	34.19 / 0.9044 / 0.1247	35.28 / 0.9291 / 0.0812	29.41	15.65
LF based	SAS (TIP 2018)	32.87 / 0.9182 / 0.0901	29.70 / 0.8680 / 0.2026	29.36 / 0.8137 / 0.2351	30.64 / 0.8666 / 0.1759	4.89	**1.35**
	EPIT (ICCV 2023)	40.58 / 0.9724 / 0.0091	32.94 / 0.9204 / 0.0603	32.98 / 0.8636 / 0.1462	35.50 / 0.9188 / 0.0719	3.28	3.00
	MGPD (TIP 2023)	38.42 / 0.9653 / 0.0129	31.85 / 0.8903 / 0.0690	31.56 / 0.8389 / 0.1676	33.94 / 0.8982 / 0.0832	14.91	25.86
	MDeRainNet (TOMM 2026)	39.40 / 0.9712 / 0.0121	34.29 / 0.9295 / 0.0451	34.03 / 0.9020 / 0.1293	35.91 / 0.9342 / 0.0622	315.91	19.04
	CGRNet (Ours)	**41.33 / 0.9849 / 0.0062**	**34.79 / 0.9362 / 0.0438**	**34.50 / 0.9107 / 0.0943**	**36.87 / 0.9439 / 0.0481**	3.57	3.31

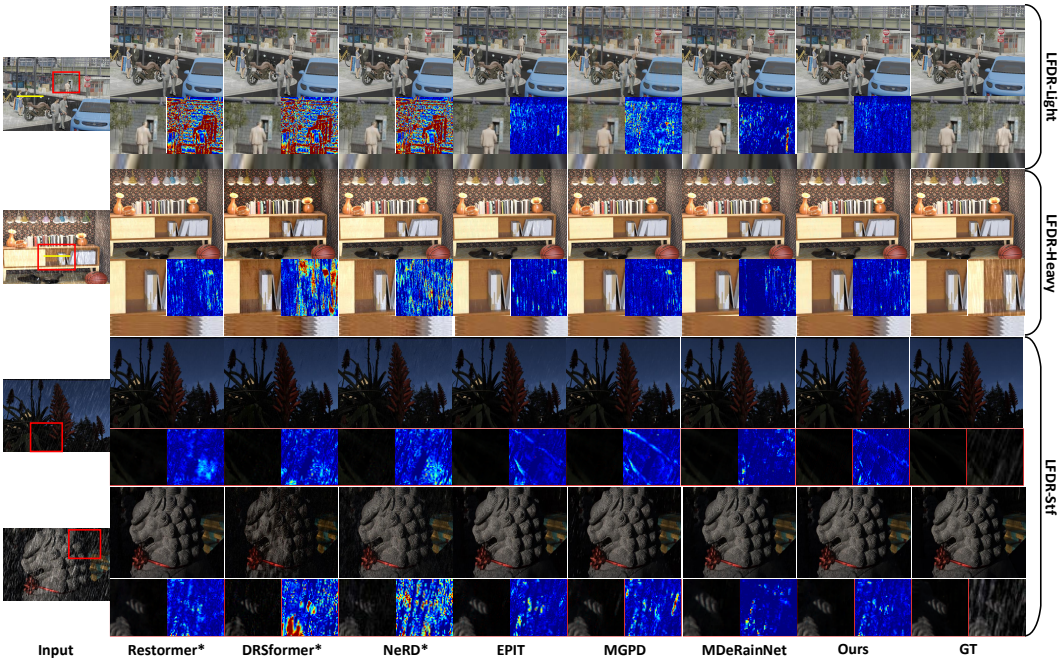

Figure C: Visual comparison on the *LFDR-Light*, *LFDR-Heavy* and *LFDR-STF*. The central view and EPIs of the recovered LFs are shown. In the error map, red indicates a large discrepancy between the output and the GT. Our method restores more details.

into pseudo-video sequences and retrain them on our datasets. Due to the structural differences between video sequences and LFs, video-based methods fail to achieve optimal performance. As shown in Table A, our CGRNet significantly outperforms them, yielding the highest average PSNR of 36.87 dB, which exceeds the best-performing video baseline (ESTINet) by a clear margin of 1.59 dB.

Fig. C provides further qualitative comparisons of our model against state-of-the-art methods on the *LFDR-Light*, *LFDR-Heavy*, and *LFDR-STF* datasets. As clearly observed in the error maps, our method yields significantly fewer regions of high discrepancy (indicated by red highlights) compared to the competing approaches, demonstrating a substantially lower error rate and closer alignment with the GT. Furthermore, the resulting EPIs generated by our method exhibit sharper and more continuous linear structures. Crucially, as shown in the depth map visualizations(Fig. E), competing methods often struggle to differentiate rain streaks from the scene structure, leading to noisy and inaccurate depth estimation. In contrast, our approach generates cleaner and more accurate depth maps, effectively isolating the background geometry from the complex rain layers. This indicates that our approach not only effectively removes rain streaks but also successfully preserves the essential epipolar consistency unique to LFs, thereby accurately restoring the underlying geometric structures and disparity information of the scenes.

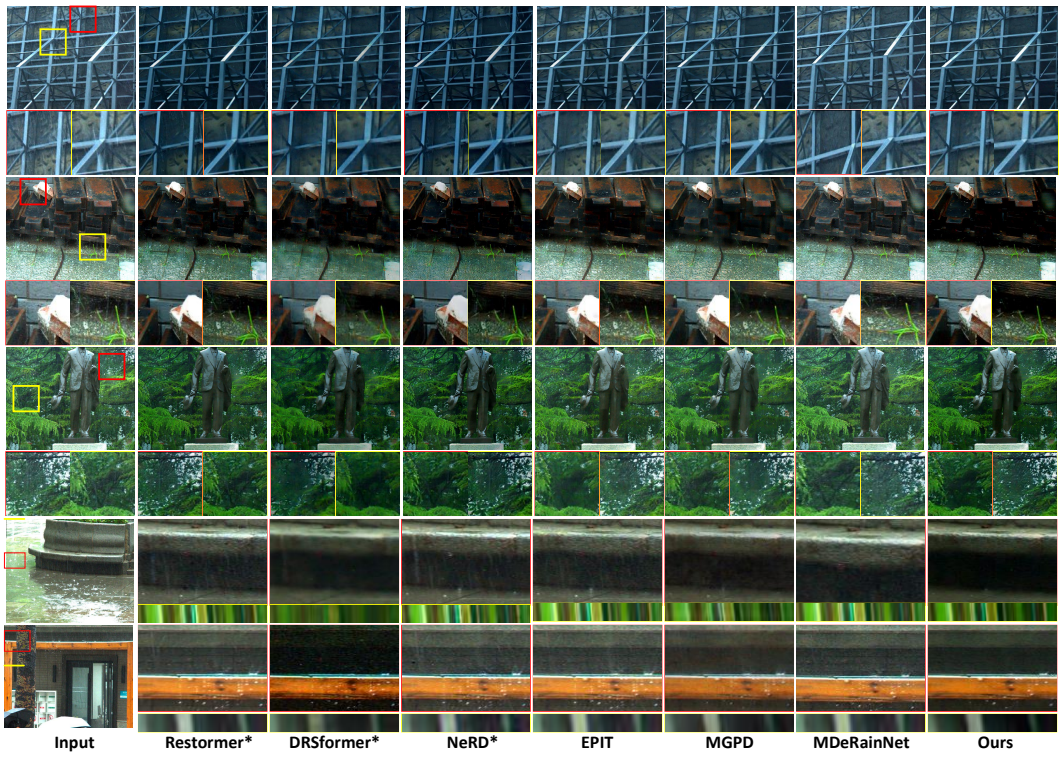

Figure D: Visual comparison on the *LFRR*, where the central view of the recovered LFs are shown. Details are magnified for clearer comparison. The results recovered by our model are clearer and more realistic, while other methods retain visible rain streaks and exhibit strong artifacts.

C.2 Comparison on Real Scenes

Fig. D present further qualitative comparisons on the real-world LFRR-Light and LFRR-Heavy datasets, respectively As demonstrated in the magnified details, competing methods struggle to handle these complexities. Specifically, other approaches either fail to completely remove the rain streaks, leaving noticeable visible residues, or introduce severe artifacts that degrade the underlying image content. Conversely, our method successfully eliminates complex rain streaks while preserving critical texture and structural details. Furthermore, the EPIs generated by our approach exhibit smooth and continuous linear patterns, confirming that our model robustly maintains spatial-angular consistency even in challenging, real-world adverse weather conditions.

C.3 Computational Efficiency

We also compare the efficiency of our methods in Table A. All execution times are evaluated on a single NVIDIA A5000 GPU, measuring the duration to restore 5×5 views. As shown in Table A, our method achieves a highly competitive parameter scale (3.57M). Although S2VD possesses fewer parameters (0.50M), our method outperforms it by a substantial margin of 4.32 dB in average PSNR, while being nearly twice as fast (3.31s vs. 6.59s). Similarly, while the pure CNN-based SAS exhibits the shortest inference time (1.35s), it suffers from severe performance degradation, lagging behind our method by over 6 dB. Furthermore, compared to our base architecture, EPIT, the calculation of our consistency map only introduces a marginal overhead of 0.29M parameters and 0.31s in runtime. Nevertheless, this slight increase yields a significant performance boost of 1.37 dB in average PSNR, demonstrating the exceptional cost-effectiveness and efficiency of our design.

C.4 Robustness of Consistency Map

In Table B, we further evaluate the robustness of our method across challenging regions. Unlike rain-induced degradations that disrupt cross-view consistency, legitimate scene structures generally

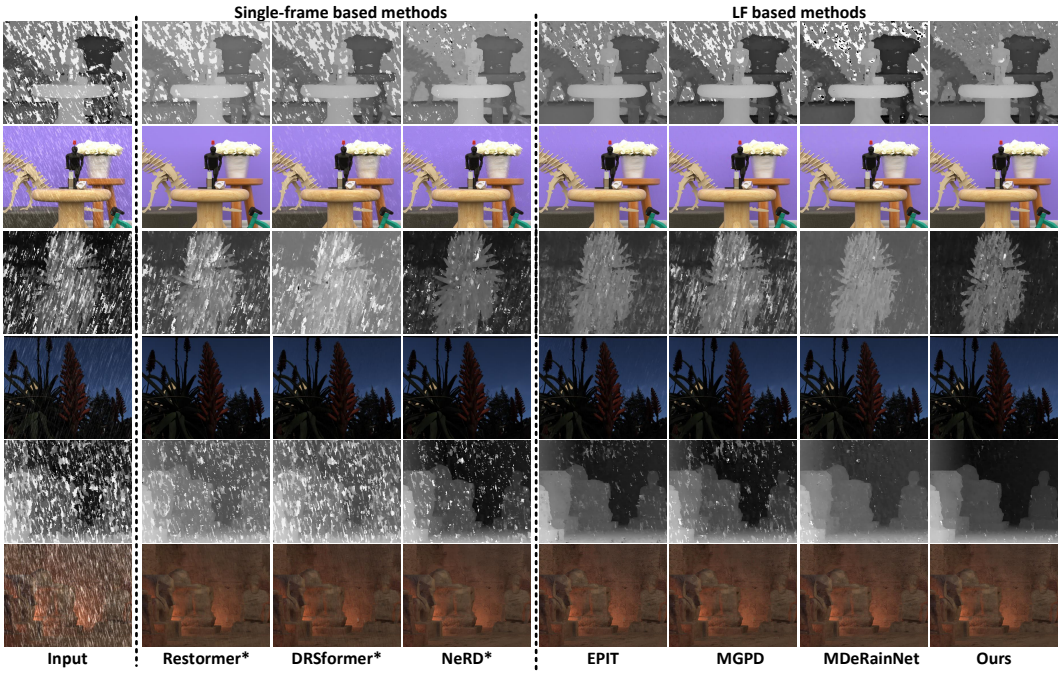

Figure E: Visual comparison of images recovered by different methods on depth estimation. The recovered images and corresponding depth maps are shown. While single-frame-based methods remove rain and show high visual quality, the depth maps are still estimated with heavy noises. In contrast, our approach produces clearer and more accurate depth maps.

Table B: Region-wise evaluation of consistency map estimation and image restoration.

Task	**Method**	Glass	Discont.	Fine struct.	Fine surr.	Overall
AP↑	JORDER	0.486	0.541	0.524	0.498	0.512
	Ours	**0.672**	**0.718**	**0.704**	**0.681**	**0.735**
PSNR↑	MDeRainNet	31.84	32.48	31.97	33.08	34.29
	Ours	**32.76**	**33.31**	**32.86**	**33.72**	**34.79**

Note: Glass: non-Lambertian; Discont.: occlusion boundaries; Fine struct.: complex textures; Fine surr.: adjacent regions.

retain reliable correspondences across a subset of views, even in the presence of occlusions or specular reflections. This distinction enables our method to better identify rain-affected regions while preserving underlying scene structures. Quantitatively, our method consistently achieves higher AP than JORDER in consistency map estimation. Benefiting from its robustness in these challenging regions, our method also achieves superior image restoration performance compared with MDeRainNet. Specifically, it improves PSNR by 0.92 dB, 0.83 dB, and 0.89 dB on Glass, Discontinuities, and Fine structures, respectively, exceeding the overall improvement of 0.50 dB.

C.5 Visualization of consistency map

The consistency maps produced by our method on both synthetic and real scenes are visualized in Fig. F. As shown, our consistency maps effectively distinguish the degraded areas, where areas affected by rain streaks exhibit notably low consistency scores.

C.6 Ablation Study

Fig. G shows the related results of the variant models. It is clear that the results restored by model-7 still contain residual rain streaks, especially under heavy rain conditions. By incorporating the

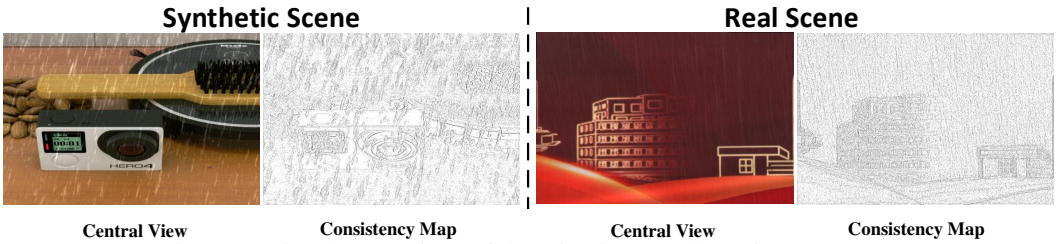

Figure F: Visual comparison of the rainy input and consistency maps.

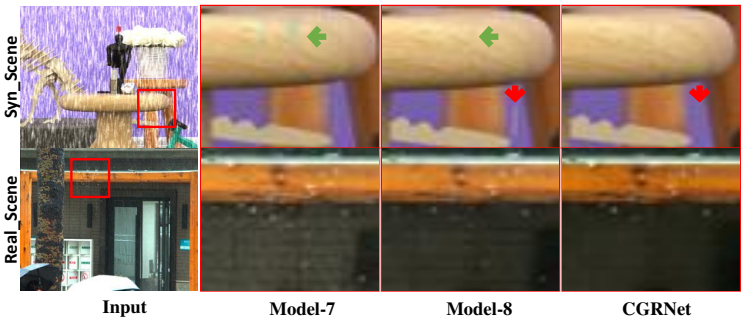

Figure G: Visual comparison of images recovered by the variant models.

refinement module (Model-8), the information sampled across the 4D space effectively reconstructs the missing details, thereby validating the necessity of the 4D refinement operation.

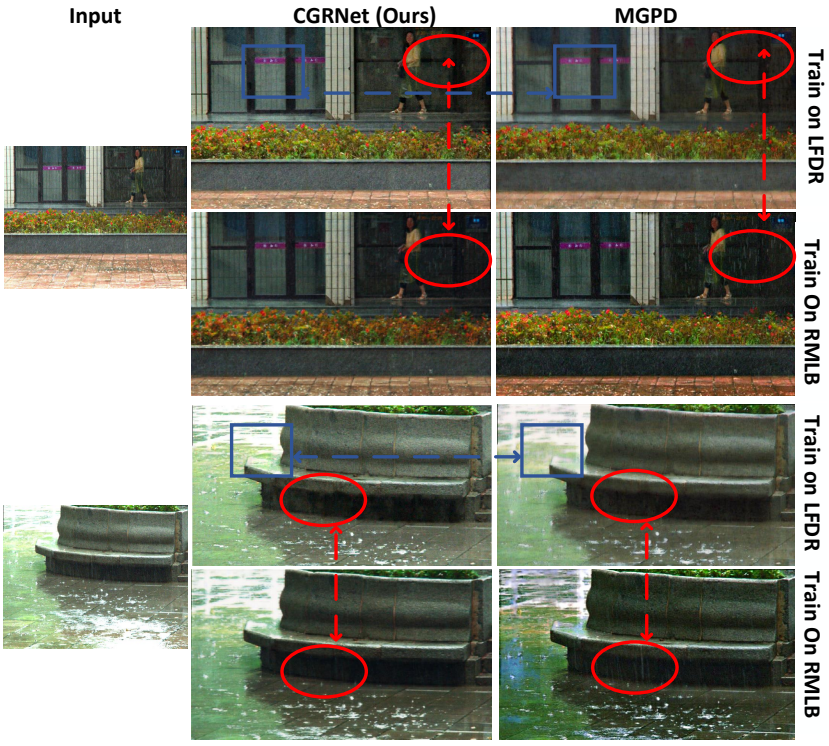

Figure H: Visual comparison of models trained on different datasets.

C.7 Dataset Validation

Fig. H illustrates the performance of models trained on different synthetic datasets when evaluated on real-world scenes. It is clear that models trained using our *LFDR* dataset produce cleaner images with

more details than those trained on *RLMB*. Besides, the color of our recovered results is noticeably more realistic without any noticeable color bias. This demonstrates that our dataset better characterizes real rain scenes and significantly improves model performance on real images.

D Limitations

Although our proposed dataset and method achieve promising results in the deraining task, our current dataset synthesis pipeline relies on pre-estimated depth maps of clean backgrounds to ensure accurate occlusion relationships. Such a requirement restricts its broader application in real-world scenarios, where obtaining accurate depth information is often costly and impractical. To reduce this dependency, our future work will explore robust, depth-free synthesis pipelines. Furthermore, beyond LF data, our core mechanism can be readily transferred to general multi-view scenarios. We plan to extend our framework to sparse and unstructured multi-view deraining tasks by effectively leveraging the available epipolar information.

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: see Sec. 1 (Introduction), where we explicitly summarize our main contributions, which are consistently supported by the theoretical derivations and experimental results in the following sections.

   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 explicitly discuss the limitations of our proposed method, including the reliance on depth maps in dataset synthesis, and outline our future directions toward depth-free and generalized multi-view deraining. The details are provided in Appendix. D.

   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: We clearly state all assumptions and provide the complete proofs for our theoretical derivations (e.g., the theoretical foundation of our consistency map) in Section 4, while the corresponding dataset validation is presented in the Section 5 and 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: We fully disclose the necessary information to reproduce our main experimental results. The proposed network architecture, loss functions, dataset usage, and implementation details (e.g., hyperparameters and optimization strategies) are comprehensively described in the main paper.

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: We plan to release the full source code, pre-trained models, and instructions upon acceptance or publication of the paper. We have provided comprehensive implementation details in the paper to aid reproducibility.

   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: We specify the training and testing details, including datasets, data splits, hyperparameters, and the optimizer, in the Implementation Details section of the main paper.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Following the standard evaluation protocol in low-level vision and LF image deraining, we report the average quantitative metrics (e.g., PSNR and SSIM) over widely used, fixed benchmark datasets. Since the evaluation process on these test sets is deterministic, error bars or statistical significance tests are typically not reported.

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: We detail the computing infrastructure, including the specific GPU hardware (e.g., NVIDIA RTX A5000) and software framework used to run our experiments, in the Implementation Details section (Sec. 4.3). We also report the training time and computational costs in Experiment section C.3 .

   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: The research conducted in this paper conforms in every respect with the NeurIPS Code of Ethics.

   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: [N/A]

    Justification: Our work focuses on the foundational low-level vision task of light field image deraining. It aims to improve image quality and does not have a direct path to negative societal impacts. Therefore, we do not explicitly discuss broader societal impacts in the paper.

    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: Our paper focuses on light field image deraining, which does not pose a high risk for misuse.

    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:[Yes]

Justification :We properly cite the original papers for all existing datasets (e.g., the HCI dataset) and baseline methods used in our experiments(Sec. 5). The details and licenses of these assets are respected.

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: While the paper introduces a new dataset and network architecture, the actual assets (e.g., data files and source code) are not released at the time of initial submission. They will be made publicly available upon acceptance. The detailed documentation and validation of the dataset are provided in the main paper and Appendix.

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: [Yes]

Justification: We conducted a randomized and anonymized human perceptual study involving 220 participants to evaluate the visual quality of restoration results on 30 real-world light field rainy scenes. Participants compared the results according to five perceptual criteria: rain removal effectiveness, detail and structure preservation, color and illumination fidelity, artifact suppression, and cross-view consistency. The presentation order was randomized to reduce positional bias. The study procedure and evaluation criteria are described in Sec. 5 and the corresponding experimental results are reported in Table 3. No personally identifiable information was collected.

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: [Yes]

    Justification: The human perceptual study involved only anonymous and low-risk visual quality assessment. No personally identifiable, sensitive, medical, or behavioral information was collected, and participants were only asked to compare image restoration results. According to the applicable institutional policy, this type of anonymous perceptual evaluation does not require formal IRB approval. Participants were informed of the study purpose and procedure before participation, and no foreseeable risks beyond ordinary image viewing were involved.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    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.