FlareX: A Physics-Informed Dataset for Lens Flare Removal via 2D Synthesis and 3D Rendering

Lishen Qu[1,3], **Zhihao Liu**[3], **Jinshan Pan**[4],
Shihao Zhou[1,3], **Jinglei Shi**[3,5], **Duosheng Chen**[3], **Jufeng Yang**[1,2,3,*]
[1]Nankai International Advanced Research Institute (SHENZHEN·FUTIAN)
[2]Peng Cheng Laboratory [3]College of Computer Science, Nankai University
[4]Nanjing University of Science and Technology
[5]Key Lab of SCCI, Dalian University of Technology
{qulishen, 2212602, zhoushihao96, duoshengchen}@mail.nankai.edu.cn
jspan@njust.edu.cn, {jinglei.shi, yangjufeng}@nankai.edu.cn
https://github.com/qulishen/FlareX

Abstract

Lens flare occurs when shooting towards strong light sources, significantly degrading the visual quality of images. Due to the difficulty in capturing flare-corrupted and flare-free image pairs in the real world, existing datasets are typically synthesized in 2D by overlaying artificial flare templates onto background images. However, the lack of flare diversity in templates and the neglect of physical principles in the synthesis process hinder models trained on these datasets from generalizing well to real-world scenarios. To address these challenges, we propose a new physics-informed method for flare data generation, which consists of three stages: parameterized template creation, the laws of illumination-aware 2D synthesis, and physical engine-based 3D rendering, which finally gives us a mi**X**ed flare dataset that incorporates both 2D and 3D perspectives, namely **FlareX**. This dataset offers 9,500 2D templates derived from 95 flare patterns and 3,000 flare image pairs rendered from 60 3D scenes. Furthermore, we design a masking approach to obtain real-world flare-free images from their corrupted counterparts to measure the performance of the model on real-world images. Extensive experiments demonstrate the effectiveness of our method and dataset.

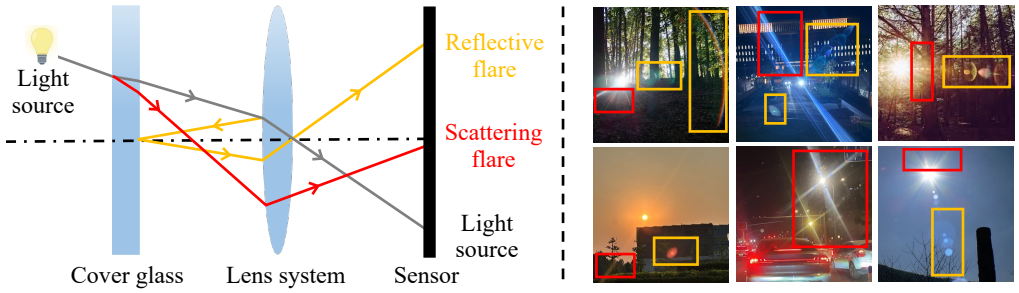
1 Introduction

Lens flare often appears when capturing images against strong light, due to light scattering and reflection within the lens system [1, 2]. This phenomenon leads to color shifts and loss of information [3, 4], resulting in degraded image quality and lowering the performance of downstream vision tasks [5, 6, 7]. Depending on various lens designs, as shown in Figure 1(a) lighting conditions, flares can differ greatly in color, shape, intensity, and spatial extent as depicted in Figure 1(b), making flare removal a challenging task.

Traditional methods [8, 9] add an anti-reflective coating to the lenses to reduce reflective flares. However, the coating's high cost and limited effectiveness restrict its widespread adoption in consumer-grade devices. To benefit from lower costs and stronger flare removal capability, researchers [10, 11, 12] begin to remove flare through deep learning algorithms [13, 14, 15, 16, 17, 18, 19, 20], driven by the large dataset [3, 21]. Besides, it is challenging to obtain a large number of paired flare images through shooting. Cleaning the lens may suppress scattering flare, but cannot eliminate reflective flare caused by lens internal imperfections [22] and intense flare.

*Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.



(a) Flare formation model (b) Real-world flare-corrupted images

Figure 1: Flare formation mechanisms and real-world flare-corrupted images. (a) The red and yellow lines respectively represent light rays that create scattering and reflective flares. (b) We use the same colors as (a) to highlight the corresponding flares. Under various lighting conditions and lens designs (e.g., aperture size, lens material), flares appear in a wide range of colors and shapes (e.g., circles, polygons, and streaks).

Given the difficulty of obtaining paired flare images in the real world, previous works [1, 2, 23] rely on simulations. Wu *et al.* [24] and Dai *et al.* [3] build datasets of flare templates using Spread Point Function and Adobe After Effects, respectively. Then, they added these templates to the background image [25] to synthesize flare-corrupted images. These datasets [24, 3] include flare patterns caused by point light sources but lack coverage of real-world scenarios with multiple reflections and non-spherical light sources. Besides, their synthesis process neglects the physical properties of flares. For example, the overall brightness of the flare is closely correlated with the distance between the light source and the camera. As a result, models trained on these datasets have limited effectiveness in coping with various types of flares in the real world.

To train a robust neural network model, it is necessary to have a realistic dataset with rich flare patterns. In this paper, we present a data generation framework based on a 3D physical engine Blender, which consists of three stages, as shown in Figure 2. First, to better simulate real-world lens flares, we parameterize key factors, such as light intensity, the times of reflection, and glass pollution. This enables us to generate 9,500 flare templates derived from 95 types of flare, covering a wider range of patterns in the real world, compared to previous datasets [24, 3]. Second, in the synthesis process, we incorporate *the laws of illumination* [26, 27, 28] to establish the relationship between the intensity of the flare and the spatial position of the light source. With the help of an estimated depth map, the intensity of flares in the synthetic images appears more realistic than random addition. Finally, we construct 60 3D scenes by customizing the placement of flares in the appropriate place instead of adding them randomly. Leveraging the 3D physics engine, we render 3,000 flare image pairs that inherently follow physical laws. By mixing 2D synthetic image pairs derived from templates and rendered images from various perspectives in 3D scenes, we introduce a physics-informed dataset called FlareX, which can support future research in flare removal. In addition, due to existing methods [24, 3] that struggle to capture the image without flares, we propose a masking approach to evaluate the model's performance in the real world. In this way, we collect 100 image pairs with a resolution of 3024×3024, containing various types of flare for testing.

Our main contributions can be summarized as follows: (i) To address the limitations of flare pattern diversity in existing datasets, we create 9,500 templates derived from 95 types of flare with different parameter settings. (ii) We improve the existing 2D synthesis pipeline by incorporating the laws of illumination. And we further build 60 3D scenes with flares to render 3,000 image pairs as complements to build a physical-realistic dataset. (iii) We propose a masking approach to obtain flare-free images from flare-corrupted ones for better model assessment, and carry out extensive experiments to demonstrate the effectiveness of our method and dataset.

2 Related work

Flare training dataset. Due to various constraints and variables, collecting a large-scale dataset of paired flare images in the real world is an extremely challenging task [30]. Wu *et al.* [24] divide a flare-corrupted image into a flare template and a background, thereby building the first semi-synthetic flare dataset, which includes 2,001 captured flare templates and 3,000 simulated flare templates.

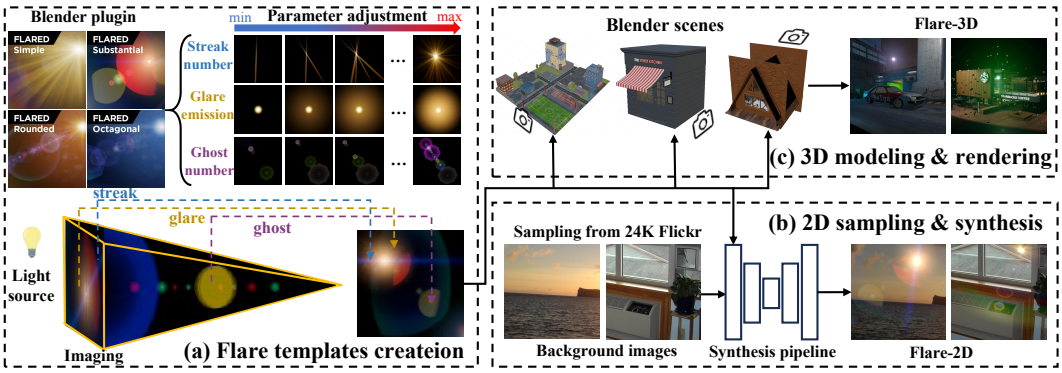


Figure 2: Illustration of our flare dataset generation framework. (a) The raw material of flares comes from the Blender plugin [29]. We manually create flare templates and adjust both flare and camera parameters to generate a wide variety of templates. (b) The synthesis pipeline generates flare-corr upted images by adding flare templates to background images sampled from the 24K Flickr image dataset [25]. (c) The 3D rendering approach constructs scenes with flare and renders them from various camera perspectives.

However, given the insufficient variety of Wu's dataset and inadequate consideration of nighttime conditions, Dai *et al.* [3] use the plugin in Adobe After Effects (AE) to create nighttime flares, including 5,000 scattering flare templates and 2,000 reflective flare templates, called Flare7K dataset. Then, Dai *et al.* capture 962 flare templates using three different cameras, expanding this collection into the Flare7K++ dataset [21]. Recently, Florin *et al.* [31] develop a synthetic flare dataset named SDFRD, specifically targeting digital single-lens reflex cameras.

Flare-corrupted image generation. The current methods randomly add the flare templates of [24, 3, 21] to the background images which are sampled from the 24K Flickr [25]. Wu *et al.* [24] and Dai *et al.* [3] apply random affine transformations to flare templates without considering physical principles, resulting in noticeable unrealism (e.g., excessive brightness or oversized flares). Zhou *et al.* [32] argue that simple addition and numerical truncation can cause overflow and distribution shift, leading them to propose an improved synthesis method. Jin *et al.* [33] demonstrate that lens flares captured by the same imaging system tend to be similar. Consequently, they select the same flare templates when randomly adding multiple flares which yields better results. To address the issue of unrealistic brightness, we improve the method of overlaying flares onto background images by introducing the laws of illumination. To position flares in more appropriate locations instead of placing them randomly, we further render 3D scenes to generate the remainder of the dataset.

Flare test dataset. Wu *et al.* [24] provide a test set comprising only 20 real image pairs, making it insufficient for evaluating performance in the real world. Dai *et al.* [3] extend this by introducing a nighttime test set containing 100 real captured flare image pairs. However, both datasets suffer from relatively low resolution (typically 512×512) and mainly contain simple and small-scale flare patterns. FlareReal600 [34] improves the resolution by providing 50 image pairs at 2K, but the ground truth is still acquired by cleaning the lens, which cannot eliminate internal reflective and severe flares. Zhou *et al.* [32] address the device diversity issue by collecting a test set using 10 mobile phones, but their dataset lacks ground truth altogether, making quantitative evaluation impossible. To tackle the difficulty of capturing reflective flare image pairs in real-world conditions, BracketFlare [22] constructs the first synthetic dataset specifically for reflective flares. Overall, current flare test datasets are limited either by resolution, diversity of flare types, or the lack of reliable ground truth, motivating the need for a more comprehensive evaluation benchmark.

3 FlareX dataset

Our dataset creation process consists of three stages: creating flare templates, 2D synthesis using the templates, and directly rendering 3D scenes. By incorporating the laws of illumination into the 2D synthesis process, we generate a large number of flare images with realistic intensity. Furthermore,

(a) Background (b) Wu *et al.* (c) Flare7K (d) Flare7K++ (e) Ours (f) Our templates

Figure 3: Comparison of synthetic flare-corrupted images and examples of our flare templates. Featuring various light sources and flares in different colors and shapes, our flare templates are capable of simulating sun halo effects in daytime and flares produced by non-spherical light sources. Note that Flare7K++ [21] specifically refers to the additional flare templates beyond those in Flare7K [3].

Table 1: Comparison of existing flare datasets. "✗" and "✓" indicate whether the dataset has the property. Note that the dataset of Wu *et al.* [24] only contains daytime flares while Flare7K [3] and FlareReal600 [34] only include nighttime flares.

Dataset	Type	Number	Multiple reflection	Light source annotation	Day	Night
Wu *et al.* [24]	3	5001	✗	✗	✓	✗
Flare7K [3]	35	7000	✗	✓	✗	✓
FlareReal600 [34]	-	600	✗	✗	✗	✓
Flare-R [21]	-	962	✗	✓	✗	✓
SDFRD [35]	3	-	✗	✗	✓	✓
FlareX (Ours)	95	9500+3000	✓	✓	✓	✓

3D scene rendering naturally adheres to physical laws, and we can place flares in locations where they are more likely to appear, rather than randomly adding them to backgrounds. To leverage the strengths of both 2D synthesis and 3D rendering, we mix the data generated by them to propose a physics-informed flare dataset, namely FlareX. We compare our dataset with existing datasets [24, 3, 34] in Table 1. Our dataset offers 9,500 flare templates derived from 95 types of flares to synthesize image pairs, along with 3,000 pairs of 3D-rendered images, featuring diverse patterns and covering both daytime and nighttime scenarios. For ease of reference, we abbreviate the two parts of FlareX as Flare-2D and Flare-3D in the following text.

3.1 Flare template creation

The existing datasets [3, 24] lack diversity and find it challenging to physically simulate lens flare, as this requires computing mutual constraints between flare components, which is a highly complex task. To solve the issues of diversity and unrealism, we use the 3D graphics engine Blender to simulate flares in alignment with physical laws.

First, we create a flare that includes multiple components, such as light source, streak, iris, and glare. We utilize the flare patterns from the Flared plugin to avoid creating flares from scratch. We manually adjust the camera focal length and various parameters of these flare components, including position, color, size, shape, etc. Most importantly, we bind the light source to a spatial point and apply the mutual constraints preset by Blender [29] among flare components, allowing the movement of the light source to produce pattern changes that more closely resemble real-world behavior. We set the background to black and allow Blender rendering in flat to produce the flare templates. After removing all flare components except the light source, we re-render the scene to obtain the corresponding light source templates. This process allows us to generate annotations for all flare components. Referring

Figure 4: 19 categories of flare templates. The 9 categories on the left represent the basic flare types, while those on the right are more complex types, officially referred to as "XT".

to real-world flares, we create a total of 95 flare types using components in Figure 4, with each type producing 100 templates by adjusting parameters, resulting in 9,500 flare templates.

In Figure 3, we present synthetic images derived from our flare templates and those of previous works [24, 3, 21], showing scenes in daylight, nighttime, and indoor settings, respectively. Templates in previous datasets feature limited flare patterns, making them ineffective for synthesizing images across diverse scenes. In both daylight and nighttime scenes, our flare templates, with multi-reflection properties and controllable parameters, create more realistic effects that naturally integrate into the environment. For the indoor scene, we show the ability of our templates to simulate flares caused by the non-spherical light source.

3.2 Flare-2D synthesis

As existing data synthesis methods [24, 3] ignore the relationships between the appearance of lens flare and their spatial position, we improve them by introducing the laws of illumination [26, 27]. Different from existing methods, our data synthesis method can synthesize flare-corrupted images whose flare intensity adheres to physical laws. Our synthesis pipeline is illustrated in Figure 5.

First, we perform multiple random affine transformations of the flare separately and estimate the depth map of the background image using a pre-trained monocular depth estimation model [36]. Second, we develop a Brightness Adjustment Module (BAM) to adjust the brightness of flares that have been affine transformed. The above operations can be expressed by:

$$\mathbf{F_i''} = BAM(\mathcal{T}_i(\mathbf{F}), \mathcal{D}(\mathbf{B})) = BAM(\mathbf{F_i'}, \mathbf{D}), \tag{1}$$

where $\mathcal{T}_i$ and $\mathcal{D}$ denote the i^{th} Random Affine Transformation and Depth Estimation, respectively. $\mathbf{F}$ denotes the origin flare image. $\mathbf{F_i'}$ is the flare image after the i^{th} random affine transformation. $\mathbf{F_i''}$ is the flare image after brightness adjustment. $\mathbf{B}$ represents the background image and $\mathbf{D}$ represents its depth map.

Finally, we obtain multiple flares after adjusting the brightness, then add them to the background image to synthesize the final image with flares, which can be represented as:

$$\mathbf{I}_{\text{sys}} = Clip(\mathbf{B} + \sum_{i=1}^{n} \mathbf{F_i''}), \tag{2}$$

where $Clip(\cdot)$ denotes clipping the addition to the range of $[0, 1]$, and n represents the number of flares to be generated.

Brightness adjustment module (BAM). Intuitively, the closer the light source is to the lens, the more intense the flare appears. The laws of illumination [26] allow for a quantitative portrayal of this physical phenomenon and the formula is as follows:

$$E = \frac{I \cdot \cos\theta}{d^2}, \tag{3}$$

where E indicates the illumination at a point on the plane, and I is the luminous intensity of the light source. θ is the angle between the optical axis and the incident light rays, and d is the distance from the light source to the illuminated point. To simplify explanations, we use the terms "angle of incidence" and "depth" in the rest of the paper.

We first perform Spatial Position Estimation (SPE) using the depth map and affine-transformed flare images to obtain the depth d_i and the angle of incidence θ_i for each affine-transformed flare image.

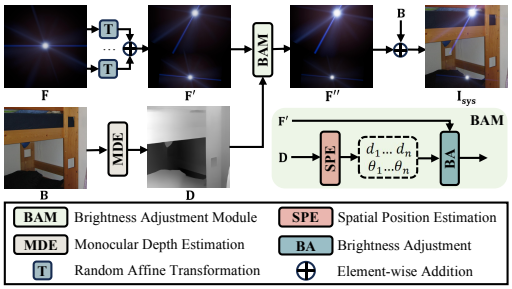


Figure 5: Our synthesis pipeline. The flare-corrupted images generated based on the laws of illumination appear more realistic.

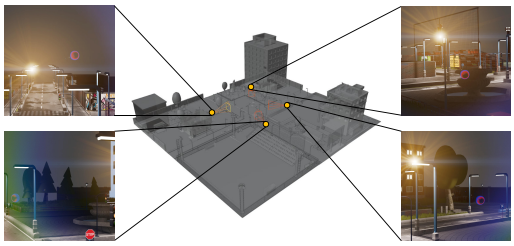


Figure 6: The 3D scene and the flare-corrupted images rendered from different camera positions. Due to the physical constraints, the variations in the shape of the flare and its position in the image align with physical laws.

In SPE, we utilize the average depth of all pixel points of the light source as the real distance between that light source and the lens, as expressed by the following equation:

$$d_i = \frac{1}{N}\sum_{j=1}^{N} \mathbf{D}_{(x_j,y_j)} \quad , \quad (x_j, y_j) \in \mathbf{L_i}, \tag{4}$$

where $\mathbf{L_i}$ symbolizes the light source area of the affine-transformed flare and N represents the total number of pixel points of the light source. x_j and y_j denote the position of the j^{th} pixel point.

For the calculation of the incident angle, since the camera parameters used for taking photos are unknown, we use the horizontal field of view to estimate. According to the law of similar triangles, we can obtain the following formula:

$$\theta_i = \arctan(\frac{2r_i}{W} \cdot \tan\frac{\varphi}{2}), \tag{5}$$

where φ represents the horizontal field of view, W denotes the width of the background image, and r_i denotes the average distance from the pixel points of the i^{th} light source to the center of the image.

Compared to the scenes being captured, the size of the lens can be ignored and represented as a point. After obtaining the depth and the incident angle, we substitute these values into Equation (3) to calculate the illumination of the lens from different light sources. The formula for making the final brightness adjustment is as follows:

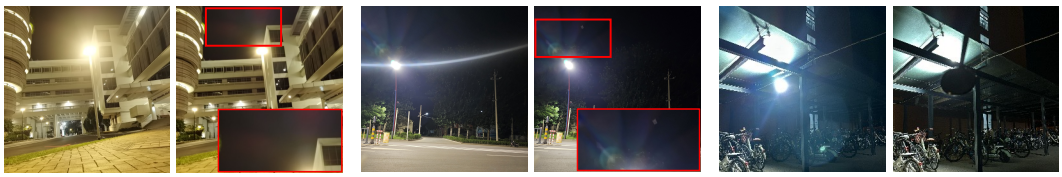
$$\mathbf{F_i''} = \mathbf{F_i'} \cdot \frac{E_i}{\frac{1}{n}\cdot\sum_{i=1}^{n} E_i} = \frac{\mathbf{F_i'}\cdot\bar{d}^2}{d_i^2 \cdot \sqrt{1+(\frac{2r_i}{W}\cdot\tan(\frac{\varphi}{2})^2)}} \quad , \tag{6}$$

where $\bar{d}$ is the average depth of the image. Specifically, we use the light with a 0° incident angle and $\bar{d}$ as a reference for adjusting the brightness of each flare. Last but not least, due to variations in the fields of view among different cameras, training various models with the same dataset may lead to poor robustness. Compared to the previous synthesis method, our method can synthesize a dataset for a specific camera by adjusting the field of view φ.

3.3 Flare-3D construction

Since the estimated depth map is not always accurate, we propose constructing flare scenarios and rendering them directly to establish more precise physical constraints. This approach naturally adheres to physical laws and produces flare-corrupted images with a more realistic intensity and appearance. The existing synthesis operations may cause image distortion due to overflow [32], which can also be mitigated by the 3D modeling method.

First, we construct a 3D scene in Blender and add lens flare into the scene. Unlike the random affine transformation method, 3D rendering allows us to customize flare placement, positioning it in areas where flares are more likely to appear, such as near light sources, in the sky, and outside windows. Next, we keyframe the camera's movement path, enabling it to follow a trajectory within the scene



(a) Flare7K [3] (b) FlareReal600 [34] (c) Ours

Figure 7: Image pairs obtained through different collection methods. The ground truths in existing test sets may retain flares, especially when capturing strong light sources.

Table 2: Quantitative results and user study evaluation. Our test set better captures the performance improvement from Flare7K [3] to Flare7K++ [21], aligning more closely with human perceptual preferences.

Dataset	Test on Flare7K [3]		Test on FlareReal600 [34]		Test on ours	
	PSNR	User study	PSNR	User study	PSNR	User study
Trained on Flare7K [3]	33%	4%	34%	8%	7%	5%
Trained on Flare7K++ [21]	67%	96%	66%	92%	93%	95%

and capture images from various perspectives, as shown in Figure 6. Compared to 2D synthetic images, the flares in rendered images exhibit different appearances related to their spatial position, making them closer to real-world images. Finally, we move the camera along the predefined path to capture flare-corrupted images and repeat the process after removing the flares to capture flare-free ones. We construct 60 scenes with different flares and render 3,000 image pairs to develop Flare-3D.

Table 3: Quantitative comparison of various image restoration models on the proposed test set. We train the same models on five datasets, and the best and second-best scores are in **bold** and underlined.

Dataset	HINet [37]			MPRNet [38]			Uformer [39]			Restormer [40]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Wu *et al.* [24]	22.927	0.642	0.139	20.937	0.571	0.147	22.252	0.645	0.143	21.798	0.653	0.145
Flare7K [3]	23.939	0.651	0.138	22.566	0.652	0.144	23.386	0.667	0.141	23.431	0.648	0.140
Flare7K++ [21]	25.172	0.678	0.140	22.604	0.612	0.146	23.615	0.667	0.142	24.495	0.671	0.138
FlareReal600 [34]	23.233	0.617	0.139	22.587	0.596	0.147	24.079	0.658	0.140	22.821	0.630	0.157
FlareX (Ours)	**25.388**	**0.682**	**0.131**	**23.882**	**0.660**	**0.138**	**25.459**	**0.692**	**0.133**	**25.096**	**0.688**	**0.131**

3.4 Real-world image collection

The test images of Zhou *et al.* [32] lack ground truths, while those from Wu *et al.* [24], Flare7K [3], and FlareReal600 [34] contain almost no reflective flares. Flare7K [3] and FlareReal600 [34] simulate typical lens dirt patterns by polluting the cover glass to capture flare-corrupted images, then capture flare-free images by cleaning the front lens. However, this method fails to eliminate strong and reflective flares in the ground truths, as shown in Figure 7, which can severely distort quantitative evaluations. In some cases, cleaner flare removal by the model even receives lower scores in the residual regions due to mismatches with flawed ground truth. Besides, it is physically impossible to obtain paired images of reflective flares [22] due to the internal reflection, so only a synthetic reflective flare test set [22] is available.

To address this challenge, we propose a masking method to collect completely flare-free images. Since flares are typically caused by intense light sources, both scattering and reflective flares can be effectively removed by simply blocking these strong lights. Specifically, after capturing the flare-corrupted image, we use an eye-exam occluder to block the direct light source and then capture the flare-free image under the same conditions. We then annotate the area where the occluder is placed, and this region is excluded from the quantitative evaluation of the performance metrics. We collect 100 pairs of test images using different smartphones and professional cameras. Our comparison with existing test sets, as shown in Table 2, highlights that our benchmark better aligns with user preferences. Nearly one-third of the samples in existing test sets fail to accurately reflect the performance improvement from Flare7K [3] to Flare7K++ [21]. Additionally, off-screen flare captured by flagship smartphones with large apertures has sparked widespread online discussions. Off-screen flare appears when light enters the camera from specific angles, presenting as thick, band-like artifacts, which can not be effectively removed by existing methods. We collect 63 representative

(a) Input (b) Wu *et al.* [24] (c) Flare7K [3] (d) Flare7K++ [21] (e) Ours (f) GT

Figure 8: Visual comparison of flare removal on the proposed test set. The name of each column represents the dataset that is used to train Uformer. Compared to (b), (c), and (d), (e) shows the best flare removal performance.

(a) Input (b) Wu *et al.* [24] (c) Flare7K [3] (d) Flare7K++ [21] (d) Ours

Figure 9: Visual comparison of off-screen flare removal. Previous methods struggle to remove these wide, white streak-like flares, whereas our approach successfully eliminates them.

images captured by various smartphones from the Internet. Our proposed masking method and collected dataset support a comprehensive evaluation of model performance in lens flare removal, particularly in assessing severe, reflective, and off-screen flare removal.

4 Experiments

4.1 Comparisons with previous datasets.

Experiments setting. To ensure a fair comparison, we apply the same data aggregation approach used in previous works [3, 41]. The loss functions in our work align with the previous works [3, 21, 42], comprising the L_1 loss, the perceptual loss with a pre-trained VGG-19 [43] and the reconstruction loss. We train the same models used in the previous benchmark [3], including HINet [37], MPRNet [38], Restormer [40], and Uformer [39], with the addition of AST [44]. We perform model training on two NVIDIA GeForce RTX 3090 GPUs with 24GB of memory. The models are trained on flare-corrupted images cropped to 512×512 resolution, with a batch size of 2, for 30,000 iterations.

Table 4: The results of off-screen flare removal.

Metrics	Input	Wu *et al.* [24]	Flare7K [3]	Flare7K++ [21]	FlareX (Ours)
NIQE ↓	4.144	<u>4.023</u>	4.050	4.026	**3.967**
BRISQUE ↓	31.894	<u>29.070</u>	30.621	29.766	**28.354**

Quantitative comparison. We adopt full-reference metrics PSNR and SSIM [45] to evaluate the performance of the models trained on different datasets. Since the flare removal is a highly perceptual task, we also use the LPIPS distance [46]. These metrics on our proposed test set measure the flare removal performance in non-occluded regions. Due to off-screen flare images having no ground truths, we use NIQE [47] and BRISQUE [48] as no-reference assessment metrics.

Table 5: The ablation study of the laws of illumination. "✓" and "✗" indicate whether the laws of illumination are incorporated into the data synthesis pipeline.

Dataset	Laws of illumination	PSNR↑	SSIM↑	LPIPS↓
Flare7K [3]	✗	23.386	0.667	0.141
	✓	23.711	0.675	0.138
FlareX (Ours)	✗	25.122	0.681	0.135
	✓	25.459	0.692	0.133

(a) Input (b) w/o laws (c) w/ laws

Figure 10: Visual comparison without and with the laws of illumination. The latter can effectively remove subtle flares and avoid incorrectly removing the light source in the mirror.

Table 6: The ablation study of our dataset. Training on the entire FlareX yields the best results, while training on Flare-3D alone leads to poor performance due to its limited amount.

Method	Flare-2D	Flare-3D	PSNR↑	SSIM↑	LPIPS↓
Uformer [39]	✓	✗	24.853	0.690	0.140
	✗	✓	23.647	0.687	0.137
	✓	✓	**25.459**	**0.692**	**0.133**
Restormer [40]	✓	✗	24.457	0.616	0.139
	✗	✓	23.728	0.665	0.141
	✓	✓	**25.096**	**0.688**	**0.131**

(a) Input (b) Flare7K++ [21] (c) Ours

Figure 11: Typical failure case of flare removal. Our method may leave noticeable artifacts when addressing images with extremely intense light sources and large flare areas.

As shown in Table 3, five baselines trained on our dataset outperform the same models trained on existing datasets. Notably, Uformer surpasses the second-best one by nearly 1.38 dB in PSNR, achieves a 3.75% increase in SSIM, and reduces LPIPS by 5%, demonstrating a significant improvement. Perceptual quality assessment results of off-screen flare removal are presented in Table 4. Uformer trained on our dataset achieves the lowest NIQE and BRISQUE scores. These results demonstrate the effectiveness of our method.

Qualitative comparison. To ensure fairness, we use the same Uformer trained on different datasets to conduct visual comparisons. As shown in Figure 8, the first and third rows of images illustrate that the model trained on our dataset is capable of removing reflective flare with special patterns. The third row also highlights that the model trained on the FlareX dataset excels in eliminating large-scale and intense flare patterns, outperforming previous approaches in terms of both precision and overall effectiveness. Additionally, when it comes to off-screen flare, the Uformer [39] model, trained on our comprehensive dataset, achieves superior performance compared to other methods, as illustrated in Figure 9. This indicates that our dataset plays a crucial role in enhancing the model's ability to generalize across different flare types.

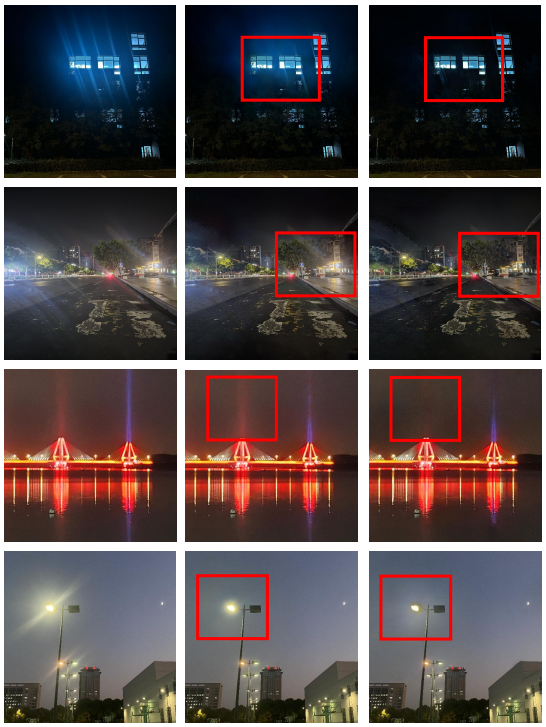
4.2 Abalation study

The laws of illumination. To demonstrate the effectiveness of our synthesis method, we carry out an ablation study to assess the influence of incorporating the illumination law. We train Uformer on FlareX with the previous method and another on Flare7K [3] with our proposed method. As shown in Table 5, integrating the laws of illumination in the synthesis process can improve model performance, which also holds true for the previous Flare7K dataset [3]. Besides, we conduct a visual comparison with and without the laws of illumination. As shown in Figure 10, the laws of illumination allow the model to remove flares more accurately.

The composition of our dataset. In order to ascertain whether the two parts of our dataset can be combined to train an optimal model, we conduct an ablation study using part of the dataset and the whole. As demonstrated in Table 6, both Uformer and Restormer achieve optimal performance when trained on the complete dataset.

4.3 More visual results.

We evaluate flare removal performance on images containing multiple flares, comparing Uformer models trained on Flare7K++ and our proposed FlareX dataset (see Figure 12). As shown in the first two rows of Figure 12, the model trained on FlareX demonstrates more effective removal of dense



(a) Input (b) Flare7K++ (c) Ours

Figure 12: Visual results of multi-flare removal. The test images come from ours (the first row), FlareReal600 [34] (the second row), and Zhou *et al.* [32] (the third and fourth row).

(a) Before (b) After

Figure 13: Object detection with flare-corrupted and flare-removed images. Flares can obscure objects such as bicycles and motorcycles, making them undetectable.

flares. Furthermore, in the fourth row, which presents a scenario involving both streak and glare flares, the FlareX-trained model also achieves cleaner and more precise removal.

We perform a visual comparison of object detection before and after flare removal. As illustrated in Figure 13, in the first and second rows, the flare-removed images reveal previously occluded objects, such as bicycles and motorcycles, which are not detected in the flare-corrupted images. Besides, in the third row, flares cause the model to mistakenly classify bicycles as motorcycles, whereas this issue is resolved after flare removal. This experiment indicates that lens flare can negatively impact the model, thereby presenting a significant threat to high-level applications.

5 Conclusion

In this paper, we present FlareX, a physics-informed dataset designed to advance the task of lens flare removal. First, to enhance pattern diversity, we generate 9,500 flare templates based on 95 physically flare types using a Blender plugin. Second, to improve the realism of synthetic data, we incorporate the laws of illumination into the 2D flare synthesis process, addressing the issue of unnatural brightness distributions common in existing methods. Finally, to further bridge the gap between synthetic and real-world images, we construct 60 scenes and render 3,000 paired samples using a physically-based rendering pipeline. Additionally, to enable more reliable real-world evaluation, we propose a masking strategy to collect 100 pairs of flare and flare-free images, effectively avoiding residual flare artifacts introduced by lens-wiping methods. Extensive experiments and ablation studies demonstrate the effectiveness of our dataset in improving flare removal performance and generalization across various models, paving the way for future research in this field.

Limitations. Despite the effectiveness of current models, extremely heavy flare remains a significant challenge, often resulting in visible artifacts after restoration, as shown in Figure 11. Future research may benefit from incorporating physical priors or transferring structural cues from cleaner regions to better recover severely degraded content.

Acknowledgement: This work was supported by Shenzhen Science and Technology Program (No. JCYJ20240813114229039), National Natural Science Foundation of China (Nos. 624B2072, U22B2049, 62302240), Natural Science Foundation of Tianjin (No.24JCZXJC00040), the Funding from SCCI, Dalian Univ. of Technology (No. SCCI2023YB01), and Supercomputing Center of Nankai University.

References

[1] Matthias Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. Physically-based real-time lens flare rendering. In *ACM SIGGRAPH*, 2011.

[2] Sungkil Lee and Elmar Eisemann. Practical real-time lens-flare rendering. *Computer Graphics Forum*, 32(4):1–6, 2013.

[3] Yuekun Dai, Chongyi Li, Shangchen Zhou, Ruicheng Feng, and Chen Change Loy. Flare7k: A phenomenological nighttime flare removal dataset. In *Advances in Neural Information Processing Systems*, 2022.

[4] Siqi Wu, Fei Liu, Yu Bai, Houzeng Han, Jian Wang, and Ning Zhang. Flare removal model based on sparse-uformer networks. *Entropy*, 26(8), 2024.

[5] Andreas Nussberger, Helmut Grabner, and Luc Van Gool. Robust aerial object tracking in images with lens flare. In *IEEE International Conference on Robotics and Automation*, 2015.

[6] Xiaoxing Mo, Leo Yu Zhang, Nan Sun, Wei Luo, and Shang Gao. Backdoor attack on deep neural networks in perception domain. In *International Joint Conference on Neural Networks*, 2023.

[7] Seon Jong Kang, Kyung Bong Ryu, Min Su Jeong, Seong In Jeong, and Kang Ryoung Park. Cam-frn: Class attention map-based flare removal network in frontal-viewing camera images of vehicles. *Mathematics*, 11(17), 2023.

[8] Andreas Nussberger, Helmut Grabner, and Luc Van Gool. Robust aerial object tracking from an airborne platform. *IEEE Aerospace and Electronic Systems Magazine*, 31(7):38–46, 2016.

[9] Ramesh Raskar, Amit Agrawal, Cyrus A Wilson, and Ashok Veeraraghavan. Glare aware photography: 4d ray sampling for reducing glare effects of camera lenses. In *ACM SIGGRAPH*, 2008.

[10] Allabakash Ghodesawar, Vinod Patil, Ankit Raichur, Swaroop Adrashyappanamath, Sampada Malagi, Nikhil Akalwadi, Chaitra Desai, Ramesh Ashok Tabib, Ujwala Patil, and Uma Mudenagudi. Deflare-net: Flare detection and removal network. In *International Conference on Pattern Recognition and Machine Intelligence*, 2023.

[11] Soonyong Song and Heechul Bae. Hard-negative sampling with cascaded fine-tuning network to boost flare removal performance in the nighttime images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023.

[12] Ishan Prakash, Aryan, S. Indu, and Mini Sreejeth. Image flare removal using deep convolutional generative adversarial networks. In *International Conference on Signal Processing and Integrated Networks*, 2023.

[13] Duosheng Chen, Shihao Zhou, Jinshan Pan, Jinglei Shi, Lishen Qu, and Jufeng Yang. A polarization-aided transformer for image deblurring via motion vector decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28061–28070, 2025.

[14] Xuhang Chen Yiguo Jiang, Shuqiang Wang Chi-Man Pun, and Wei Feng. Mfdnet: Multi-frequency deflare network for efficient nighttime flare removal. *arXiv preprint arXiv:2406.18079*, 2024.

[15] George Ciubotariu, Florin-Alexandru Vasluianu, Zhuyun Zhou, Nancy Mehta, Radu Timofte, Ke Wu, Long Sun, Lingshun Kong, Zhongbao Yang, Jinshan Pan, et al. Aim 2025 challenge on high fps motion deblurring: Methods and results. *arXiv preprint arXiv:2509.06793*, 2025.

[16] Yunshan Zhong, Mingbao Lin, Jingjing Xie, Yuxin Zhang, Fei Chao, and Rongrong Ji. Distribution-flexible subset quantization for post-quantizing super-resolution networks. *Science China Information Sciences*, 68(3):132108, 2025.

[17] Yifan Chen, Tong Tian, Xin Lu, Chen Li, Ruolan Zhu, Zhe Sun, and Xuelong Li. Attention-enhanced computational ghost imaging. *Science China Information Sciences*, 68(6):162104, 2025.

[18] Jiangxin Dong, Jinshan Pan, Zhongbao Yang, and Jinhui Tang. Multi-scale residual low-pass filter network for image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12345–12354, 2023.

[19] Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5886–5895, 2023.

[20] Yuang Meng, Xin Jin, Lina Lei, Chun-Le Guo, and Chongyi Li. Ultraled: Learning to see everything in ultra-high dynamic range scenes. *arXiv preprint arXiv:2510.07741*, 2025.

[21] Yuekun Dai, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Yihang Luo, and Chen Change Loy. Flare7k++: Mixing synthetic and real datasets for nighttime flare removal and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(11):7041–7055, 2024.

[22] Yuekun Dai, Yihang Luo, Shangchen Zhou, Chongyi Li, and Chen Change Loy. Nighttime smartphone reflective flare removal using optical center symmetry prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[23] Ruicheng Feng, Chongyi Li, Huaijin Chen, Shuai Li, Chen Change Loy, and Jinwei Gu. Removing diffraction image artifacts in under-display camera via dynamic skip connection network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[24] Yicheng Wu, Qiurui He, Tianfan Xue, Rahul Garg, Jiawen Chen, Ashok Veeraraghavan, and Jonathan T Barron. How to train neural networks for flare removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[25] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[26] Max Reiss. The cos 4 law of illumination. *Journal of the Optical Society of America*, 35(4):283–288, 1945.

[27] WA Ramadan, AS El-Tawargy, and HH Wahba. Optical phase retrieving of a projected object by employing a differentiation of a single pattern of two-beam interference. *Scientific Reports*, 13(1):14840, 2023.

[28] Lishen Qu, Shihao Zhou, Jinshan Pan, Jinglei Shi, Duosheng Chen, and Jufeng Yang. Harmonizing light and darkness: A symphony of prior-guided data synthesis and adaptive focus for nighttime flare removal. *arXiv preprint arXiv:2404.00313*, 2024.

[29] Blender Online Community. Blender - a 3d modelling and rendering package, 2024. `https://blender.org`.

[30] Xiang Chen, Jinshan Pan, Jiangxin Dong, and Jinhui 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.

[31] Florin Vasluianu, Zongwei Wu, and Radu Timofte. Sfnet-a spatial-frequency domain neural network for image lens flare removal. In *IEEE International Conference on Image Processing*, 2024.

[32] Yuyan Zhou, Dong Liang, Songcan Chen, Sheng-Jun Huang, Shuo Yang, and Chongyi Li. Improving lens flare removal with general-purpose pipeline and multiple light sources recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[33] Zheyan Jin, Huajun Feng, Zhihai Xu, and Yueting Chen. A data generation method for image flare removal based on similarity and centrosymmetric effect. *Photonics*, 10(10), 2023.

[34] Yuekun Dai, Dafeng Zhang, Xiaoming Li, Zongsheng Yue, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Peiqing Yang, Zhezhu Jin, Guanqun Liu, and Chen Change Loy. Mipi 2024 challenge on nighttime flare removal: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024.

[35] Florin Vasluianu, Zongwei Wu, and Radu Timofte. Sfnet - a spatial-frequency domain neural network for image lens flare removal. In *IEEE International Conference on Image Processing*, pages 1711–1717, 2024.

[36] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[37] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

[38] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[39] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[40] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[41] Dafeng Zhang, Jia Ouyang, Guanqun Liu, Xiaobing Wang, Xiangyu Kong, and Zhezhu Jin. Ff-former: Swin fourier transformer for nighttime flare removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023.

[42] Yousef Kotp and Marwan Torki. Flare-free vision: Empowering uformer with depth insights. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] Shihao Zhou, Duosheng Chen, Jinshan Pan, Jinglei Shi, and Jufeng Yang. Adapt or perish: Adaptive sparse transformer with attentive feature refinement for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[47] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012.

[48] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

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: Refer to Section 1 for details.

   Guidelines:

   - The answer NA 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 NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are discussed in the Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The assumptions and a complete proof are presented in Section 3.2.

Guidelines:

- The answer NA 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: In Section 3, we clearly describe the steps taken to make our datasets reproducible, including a detailed data acquisition pipeline and data processing methods. In Section 4, training details are clearly provided to make the results verifiable.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Our dataset and code are provided in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test details are specified at Section 4.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: Error bars or other statistical significance are unnecessary in our experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The main model parameters and computational complexity are in Section 4. More computer resources are provided in the supplementary material.

   Guidelines:

   - The answer NA 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: Every aspect of the experimental paper complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: The societal impacts are discussed in the supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This question is unrelated to the research topic of this article.

    Guidelines:

    - The answer NA 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: The existing technologies used have clear references and mentions.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The details about training and limitations for our newly-introduced dataset are described in the paper at Section 3 and Section 5 respectively. The license and other documentation are provided alongside the dataset in the supplementary materials.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: This question is unrelated to the research topic of this article.

    Guidelines:

    - The answer NA 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: Yes.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This question is unrelated to the research topic of this article.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.