# Light Source Guided Single-Image Flare Removal from Unpaired Data

Xiaotian Qiao  Gerhard P. Hancke  Rynson W.H. Lau
City University of Hong Kong

## Abstract

*Causally-taken images often suffer from flare artifacts, due to the unintended reflections and scattering of light inside the camera. However, as flares may appear in a variety of shapes, positions, and colors, detecting and removing them entirely from an image is very challenging. Existing methods rely on predefined intensity and geometry priors of flares, and may fail to distinguish the difference between light sources and flare artifacts. We observe that the conditions of the light source in the image play an important role in the resulting flares. In this paper, we present a deep framework with light source aware guidance for single-image flare removal (SIFR). In particular, we first detect the light source regions and the flare regions separately, and then remove the flare artifacts based on the light source aware guidance. By learning the underlying relationships between the two types of regions, our approach can remove different kinds of flares from the image. In addition, instead of using paired training data which are difficult to collect, we propose the first unpaired flare removal dataset and new cycle-consistency constraints to obtain more diverse examples and avoid manual annotations. Extensive experiments demonstrate that our method outperforms the baselines qualitatively and quantitatively. We also show that our model can be applied to flare effect manipulation (*e.g.*, adding or changing image flares).*

## 1. Introduction

Image flares are common and often undesirable light artifacts, caused by taking pictures of a scene with a very bright light source [1, 24]. Part of the light undergoes interreflection among the optical elements inside the camera, producing some unexpected light artifacts in the image [23]. These artifacts tend to appear more often with smartphone cameras, due to the poor anti-reflective coatings.

The presence of flare artifacts can affect the visual quality of images, and may inhibit understanding of underlying object/scene information and hamper the performances of existing vision tasks [31], *e.g.*, semantic segmentation and depth estimation. However, automatically detecting and removing them can be very challenging. Different combinations of lens properties and environment settings, including light source position, characteristics of the lens, and the camera angle to the light source, may lead to different types of flares with diverse shapes, colors, and positions.

(a) Input (b) Chabert et al. [4] (c) Wu et al. [31] (d) Ours

Figure 1. Single-image flare removal (SIFR). Given an input image with flare artifacts (pointed to by red arrows), existing methods, (b) and (c), can only handle limited flare types (*e.g.*, flare spot) without removing the entire flare or wrongly consider other regions as flares (*e.g.*, the cloud in the first row). In contrast, our light source aware model (d) can remove the flare more accurately.

Professional photographers may apply preventative measures during the image capturing process, such as optimized barrel design, lens hood, or anti-reflective coating, to help eliminate the flares. Unfortunately, these hardware solutions can hardly eliminate the entire flare artifacts [20, 22]. This is probably due to the diverse causes of image flares. They also cannot be applied to existing images already with flare artifacts. There have been a few attempts to remove these undesirable flares from images automatically by utilizing predefined intensity and geometry priors of flare artifacts [30, 4]. However, these methods can only handle limited flare types (*e.g.*, flare spot), as shown in Fig. 1(b). Recently, with the popularity of deep neural networks, a deep model [31] is proposed to learn to remove different types of flare artifacts from paired synthetic training data. Although it is shown to outperform existing traditional methods, it does not generalize well to diverse real-world images, as shown in Fig. 1(c).

From our study, we have made two observations. First, we observe that the light source plays an important role in the visual appearances of the generating flares, *e.g*., a series of streaks radiating outward from the light source to form a star-shaped effect, glare as a bright region around the light source that fades gradually, and a series of circles or rings in the image. These imply that the shape, brightness, and position of the light source encode useful cues about the appearances of the flares. Through learning such light source aware guidance, we can detect and remove flares more reliably. Second, it is not easy to collect a large-scale dataset of image flare pairs with diversity. Typically, the user needs to manually adjust the camera parameters in order to obtain a flare image, and then use preventative measures to obtain a flare-free image. Unfortunately, this setting often produces training pairs with inconsistent colors and exposures, due to the change of the environmental lighting.

Inspired by the above observations, we propose a light source guided learning framework with unpaired data for single-image flare removal (SIFR). First, we detect the light source region and the flare region separately with two branches. Given a single image as input, we propose a Light Source Detection (LSD) module to predict a light source mask, and a Flare Detection (FD) module to predict a flare mask. Second, we estimate a flare-free image based on the flare mask and the light source aware guidance using the Flare Removal (FR) module. Finally, we feed the predicted light source mask and the flare-free image through a Flare Generation (FG) module to learn the inverse mapping to reconstruct the input flare image. By imposing the cycle-consistency constraints [34] and learning the underlying relationships between the flare region and the light source region, we make the first attempt to address the flare removal problem through a unified framework that integrates both flare removal and flare generation tasks. Our approach allows training of the model with unpaired data, which are much cheaper to collect.

We conduct extensive experiments to evaluate the effectiveness of the proposed approach. The experimental results show that our approach is effective in removing the flare artifacts, compared with the baselines. We also show that our model can be applied to image editing, allowing the user to manipulate the flare effect in the image by changing the position or size of the light source.

To sum up, we make the first effort to utilize the light source information to guide the SIFR task, and propose a deep framework to learn from unpaired data with new cycle-consistency constraints. We also construct the first unpaired flare and flare-free image dataset with diverse scenarios. Extensive evaluations demonstrate the effectiveness of our method, both quantitatively and qualitatively, and the benefits of it on the flare effect manipulation task.

## 2. Related Work

**Traditional methods.** The detection and removal of image flares are important problems. A number of approaches have been proposed to measure, detect or remove flare artifacts, including hardware and software solutions. Most of the hardware solutions focus on improving the optical elements of the camera to remove flares. Based on the flare characteristics of different lens systems that have been studied [15], Boynton *et al*. [3] construct a fluid-filled camera to reduce the effects of unwanted light reflections. Raskar *et al*. [22] insert a transparency mask on top of the imaging sensor in order to use 4D ray sampling to reduce flare effects. Macleod *et al*. [18] find that replacing a circular polarizer with a neutral density filter can reduce the reflections from lens surfaces. However, the specific hardware modifications described above are not enough to remove lens flare artifacts completely. In addition, they can only reduce the flare artifacts during the capturing process, and cannot deal with existing images with flare artifacts.

To address the above limitations, several post-processing methods are proposed for detecting and removing image flare artifacts. Faulkner *et al*. [7] and Seibert *et al*. [25] propose to remove image flares by deconvolution using the measured glare spread function (GSF). Wu *et al*. [30] apply the proposed shadow extraction method to flare removal by using the rough user-supplied hints about the flare and flare-free regions. Talvala *et al*. [27] selectively composite a number of images taken by a static camera to block the light that contributes to flares. Koreban *et al*. [16] use a selective processing method on two frames based on the proposed flare formation model to mitigate the flare effect. Zhang *et al*. [32] propose a flare formation model to remove the flares in an image by decomposing the image into a scene layer and a flare layer. Other approaches [2, 4, 28] use a two-stage process to first estimate the flare spot region and then reconstruct the region via exemplar-based inpainting [5]. All the above works are based on hand-crafted features, which are limited by their assumptions, and cannot work well on diverse flare patterns in complex scenes.

**Deep learning-based methods.** Recently, deep learning-based approaches have shown great successes on a variety of low-level vision tasks, such as deraining [29], shadow removal [21, 9], and reflection removal [17, 6]. To our knowledge, there is only one deep learning-based approach for SIFR. Wu *et al*. [31] propose a U-Net architecture for removing flare artifacts. As it is based on a synthesis method to generate paired training data, it does not generalize well to real-world data. In contrast, we propose a new learning framework that learns to remove flare artifacts with unpaired data. Although previous works such as Mask-ShadowGAN [10] and RR-GAN [33] propose to train the

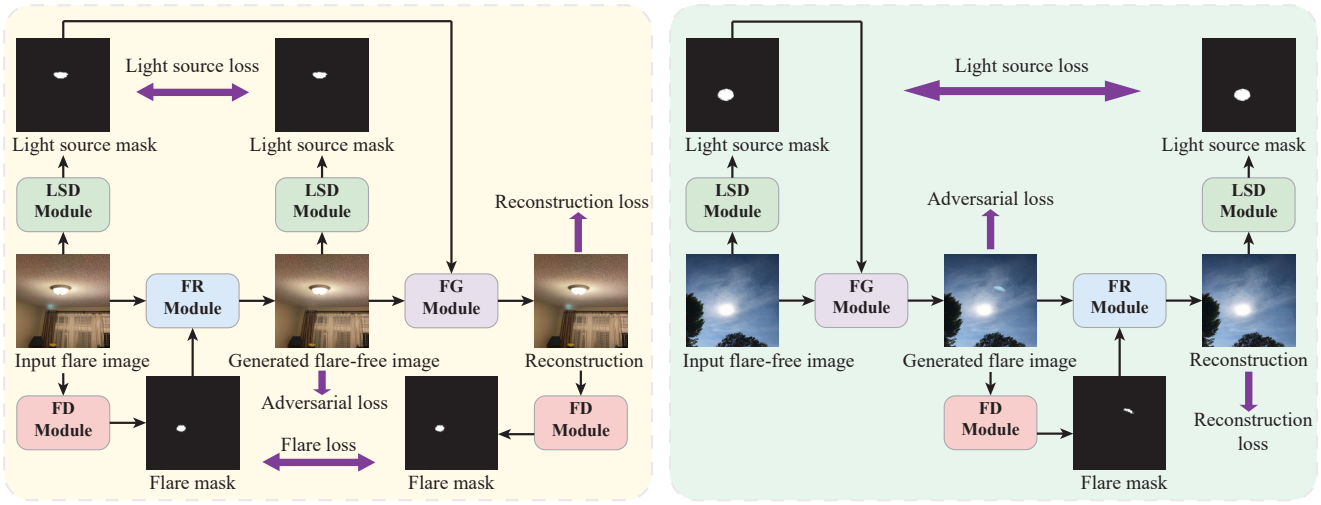


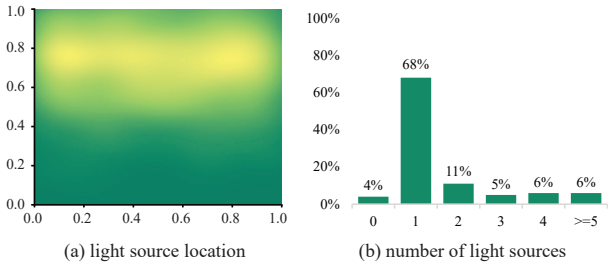

Figure 2. Our proposed light source guided learning framework for single-image flare removal. It consists of a light source detection (LSD) module, a flare detection (FD) module, a flare removal module (FR), and a flare generation (FG) module. Without the need for paired data, these modules are used to learn from either real flare images (a) or flare-free images (b). We also introduce two discriminators for the generated flare-free and flare images.

shadow removal and rain removal tasks by learning the underlying mapping between two domains with unpaired data, they only learn to predict one mask to indicate the region to be removed. Unlike the shadow removal and rain removal tasks, there is a strong distraction factor in the flare removal task, *i.e.*, the light source. Although the light source region and the flare region closely resemble each other, only the flare region should be removed. In our method, we predict both the light source mask and the flare mask simultaneously, in order to explicitly model the underlying relationships between the light source region and the flare region for SIFR, which has not been explored before.

## 3. Unpaired Flare Removal (UFR) Dataset

As far as we know, there are no publicly available flare removal datasets. Creating a large-scale dataset of image flare pairs can be very challenging. Typically, the flare and flare-free image pairs should be captured with and without flare artifacts while keeping the illumination of the scene unchanged. For each scene, we need to manually adjust the camera parameters until we get the flare artifacts to capture a flare image. We then remove these artifacts by using preventative measures to obtain a flare-free image. Such an operation is both tedious and time-consuming, making it very difficult to collect a large number of images with many types of scenarios. In addition, the training pairs will likely have inconsistent colors and exposures, due to the change of the environmental lighting or the hardware devices.

In contrast, our framework does not require paired data with pixel-wise correspondences for training. Instead, it learns only from unpaired flare and flare-free images for




(a) light source location (b) number of light sources

Figure 3. Statistics of the unpaired flare removal dataset, including the distribution of the light source location (a) and the number of light sources in the image (b).

SIFR. To increase the diversity of the images, we construct the dataset with the following guidelines:

- **Scene.** We select images that cover a variety of our daily life scenes, *e.g.*, streets, gardens, living rooms, and open spaces.
- **Light source.** The images should contain different types of light sources (*e.g.*, sun, ceiling lights, street lights, etc.), with different numbers, shapes, and at different locations.
- **Illumination.** The images should be captured under different illumination conditions, including outdoors (sunrise / sunset) and indoors (daytime / nighttime).

To create this dataset, we initially collect around 3000 flare and flare-free images from the Internet (e.g., Google and Flickr). We then invite a photographer to help filter these images by following the above guidelines. In particular, we ask the photographer to carefully check the flare-free images to make sure that they have no flares in them, as

some flares may not be easily observed. In the end, our unpaired dataset contains a total of 996 flare images and 672 flare-free images, covering a variety of real-world scenes. Figure 3 shows some statistics of our dataset. We compute the probability map to show the distribution of the light source location in the image. We can see that most of these lights appear near the top of the images.

# 4. Our Approach

Given a single flare image $I_{\text{fl}}$ as input, our goal is to learn a function $f$ that removes the flare artifacts and generates a flare-free image $I_{\text{ff}} = f(I_{\text{fl}})$. The key idea is to utilize the underlying cues from the light source to guide the SIFR task. Figure 2 shows the overall architecture of our proposed framework. It consists of a light source detection (LSD) module, a flare detection (FD) module, a flare removal (FR) module, and a flare generation (FG) module. During training, we reuse these modules to learn from either real flare images or flare-free images with the proposed constraints. During inference, we only need to use the FD module and the FR module for the SIFR task.

## 4.1. Architecture

**Light source detection (LSD) module.** The LSD module aims to detect the light source in the input image, and output a light source mask. It follows an encoder-decoder architecture. We first use an encoder to extract features from the input image by four convolution layers. We then feed the image features into a decoder to output a light source mask. It is composed of two deconvolution layers and three convolution layers.

**Flare detection (FD) module.** The FD module aims to detect the flare region in the input flare image, and output a flare mask. It has the same architecture as that of the LSD module. Both the light source mask and the flare mask are binary maps.

**Flare removal (FR) module.** The FR module aims to remove the flare artifacts in the input image, and output a flare-free image. It takes the concatenation of a flare image and the predicted flare mask as input, which has four channels in total. It is an encoder-decoder architecture with residual blocks in the middle. It includes three convolution operations, followed by nine residual blocks. The flare-free image is then generated by two deconvolution blocks and a final convolution layer. Although the light source often exists in most flare images, there are some cases that no light sources appear in them. It is worth noting that our module can still remove the flare artifacts for such cases by learning the semantic features from our dataset.

**Flare generation (FG) module.** Given a flare-free image, the FG module aims to generate flares on it to output a flare image. The input to it is the concatenation of a flare-free image and a light source mask. It has the same architecture as that of the FR module. The user can also provide a light source mask to indicate the position and shape of the light source. The FG module will then generate a flare image that corresponds to the provided light source mask.

**Discriminators.** To ensure the generation of plausible results, we introduce two additional discriminators to our network. We use a flare-free discriminator $D_{\text{ff}}$ to distinguish between the generated flare-free images and the real flare-free images, and a flare discriminator $D_{\text{fl}}$ to distinguish between the generated flare images and the real flare images. We adopt the PatchGAN [12] architecture design for both discriminators $D_{\text{ff}}$ and $D_{\text{fl}}$.

## 4.2. Learning from Flare Images

As shown in Figure 2(a), given the input flare image $I_{\text{fl}}$, the LSD module first predicts a mask $M^s_{\text{fl}}$ representing the light source region, while the FD module predicts another mask $M^f_{\text{fl}}$ indicting the flare region in the image. We then concatenate and pass the predicted flare mask $M^s_{\text{fl}}$ and the input flare image $I_{\text{fl}}$ through the FR module to obtain the flare-free image $\hat{I}_{\text{ff}}$.

Since the light source region may sometimes be similar to that of the flare region, the light source can be easily considered as a flare and removed [31]. To avoid this problem, we explicitly enforce the light source mask consistency between the input flare image and the generated flare-free image via the light source loss. In particular, we pass the input flare image and the generated flare-free image to the LSD module separately. We apply the light source loss to measure the pixel-wise difference of the two predicated light source masks:

$$L^s_{\text{fl}} = \sum \|M^s_{\text{fl}} - \hat{M}^s_{\text{ff}}\|_1, \tag{1}$$

where $M^s_{\text{fl}}$ is the light source mask from the input flare image $I_{\text{fl}}$, and $\hat{M}^s_{\text{ff}}$ is the light source mask from the generated flare-free image $\hat{I}_{\text{ff}}$.

To reduce the gap between the generated and real flare-free images, we optimize the following objective using an adversarial loss [19]:

$$L^{\text{adv}}_{\text{fl}} = E_{x\sim p_{\text{fake}}(x)}[(D_{\text{ff}}(x) - 1)^2]. \tag{2}$$

The loss for $D_{\text{ff}}$ is formulated as:

$$L_{D_{\text{ff}}} = E_{x\sim p_{\text{fake}}(x)}[(D_{\text{ff}}(x))^2] + E_{y\sim p_{\text{real}}(y)}[(D_{\text{ff}}(y) - 1)^2]. \tag{3}$$

Further, to avoid any artifacts on the generated images [12], we transform the generated flare-free image back to the input flare image and enforce the cycle-consistency constraint. Specifically, we use the mask $M^s_{\text{fl}}$ as the guidance to indicate the light source region. We concatenate $M^s_{\text{fl}}$ with the generated flare-free image $\hat{I}_{\text{ff}}$ as the input to the FG module to reconstruct the flare image $\hat{I}_{\text{fl}}$. We apply the flare

loss and the reconstruction loss to encourage the contents and the flare region to be the same. The flare loss is used to measure the pixel-wise difference between the predicted flare mask $M_{\text{fl}}^f$ from the input flare image $I_{\text{fl}}$ and the new flare mask $\hat{M}_{\text{fl}}^f$ from the reconstructed image $\hat{I}_{\text{fl}}$:

$$L_{\text{fl}}^f = \sum \|M_{\text{fl}}^f - \hat{M}_{\text{fl}}^f\|_1. \quad (4)$$

The reconstruction loss measures the difference between the prediction $\hat{I}_{\text{fl}}$ and the ground truth $I_{\text{fl}}$ in both the image and feature space:

$$L_{\text{fl}}^{\text{cycle}} = \sum \|I_{\text{fl}} - \hat{I}_{\text{fl}}\|_1 + \|\phi(I_{\text{fl}}) - \phi(\hat{I}_{\text{fl}})\|_1, \quad (5)$$

where $\phi$ is the feature map of the "conv5_3" layer of VGG-19 [26].

In summary, when learning from the flare images, the total loss $L_{\text{fl}}$ is:

$$L_{\text{fl}} = w_1 L_{\text{fl}}^s + w_2 L_{\text{fl}}^{\text{adv}} + w_3 L_{\text{fl}}^f + w_4 L_{\text{fl}}^{\text{cycle}}, \quad (6)$$

where $w_1$, $w_2$, $w_3$, and $w_4$ are the loss weights.

### 4.3. Learning from Flare-free Images

Our framework also trains in the inverse direction to learn from flare-free images. As shown in Figure 2(b), given a real flare-free image $I_{\text{ff}}$, we first use the LSD module to predict the light source mask $M_{\text{ff}}^s$. We then concatenate and pass $M_{\text{ff}}^s$ and $I_{\text{ff}}$ through the FG module to generate the flare image $\hat{I}_{\text{fl}}$. We also apply a similar adversarial loss $L_{\text{ff}}^{\text{adv}}$ as described above to optimize the FD module. This drives the generated flare images to become closer to real flare images:

$$L_{\text{ff}}^{\text{adv}} = E_{x \sim p_{\text{fake}}(x)}[(D_{\text{fl}}(x) - 1)^2]. \quad (7)$$

To enforce the cycle-consistency constraint, we use the FD module to predict the flare mask $\hat{M}_{\text{fl}}^f$ by taking the generated flare image $\hat{I}_{\text{fl}}$ as the input. We then concatenate $\hat{M}_{\text{fl}}^f$ and $\hat{I}_{\text{fl}}$ and send it to the FR module to produce the reconstructed flare-free image $\hat{I}_{\text{ff}}$. We apply the reconstruction loss $L_{\text{ff}}^{\text{cycle}}$ to force the prediction $\hat{I}_{\text{ff}}$ and the ground truth $I_{\text{ff}}$ to be the same in both the image space and the feature space:

$$L_{\text{ff}}^{\text{cycle}} = \sum \|I_{\text{ff}} - \hat{I}_{\text{ff}}\|_1 + \|\phi(I_{\text{ff}}) - \phi(\hat{I}_{\text{ff}})\|_1, \quad (8)$$

where $\phi$ is the feature map of the "conv5_3" layer of VGG-19 [26].

Further, we adopt the LSD module to produce the light source mask $\hat{M}_{\text{ff}}^s$ from $\hat{I}_{\text{ff}}$, and then use the light source loss to avoid removing the light source from the image:

$$L_{\text{ff}}^s = \sum \|M_{\text{ff}}^s - \hat{M}_{\text{ff}}^s\|_1. \quad (9)$$

In summary, when learning from the flare-free images, the total loss $L_{\text{ff}}$ is:

$$L_{\text{ff}} = w_5 L_{\text{ff}}^{\text{adv}} + w_6 L_{\text{ff}}^s + w_7 L_{\text{ff}}^{\text{cycle}}, \quad (10)$$

where $w_5$, $w_6$, and $w_7$ are the loss weights.

| | Input image | Chabert *et al.* [4] | CycleGAN [34] | Ours |
|---|---|---|---|---|
| PSNR↑ | 16.35 | 16.63 | 18.68 | **21.57** |
| SSIM↑ | 0.718 | 0.723 | 0.775 | **0.812** |

Table 1. Quantitative comparison of the proposed method with prior works. The best results are marked in bold.

### 4.4. Implementation Details

The proposed model is implemented under the PyTorch framework. Section 4.1 presents the detailed network architecture. During training, we first pre-train the modules by learning from flare-free images. In particular, we generate random light source masks as the input for the FG module and as the ground truth for light source detection. We then reuse these modules to learn from either real flare images or flare-free images with the proposed constraints. The input images are resized to $512 \times 512$ and are scaled or horizontally flipped in a random manner. The parameters of our network are randomly initialized, following a zero-mean Gaussian distribution with a standard deviation of 0.02. We adopt the Adam solver [13] to optimize the network with $\beta_1$ = 0.5 and $\beta_2$ = 0.9999. We empirically set all the loss weights to 1. We set the initial learning rate as 0.0002 and decay it by 0.1 every 100 epochs. The parameters of the FR module, the FG module, and the discriminators are alternately updated in each iteration [8]. During the testing stage, images are also resized to a resolution of $512 \times 512$ for network inference.

## 5. Experiments

In this section, we first introduce the experimental settings in Section 5.1. We then compare our results both quantitatively and qualitatively to the existing methods, and evaluate the quality of our generated flare-free images via a user study in Section 5.2. We further conduct thorough ablation studies to analyze the components of the proposed model in Section 5.3. Finally, we show how our model can be applied to flare effect manipulation in Section 5.4.

### 5.1. Experimental Settings

**Benchmark Dataset.** To evaluate the performance of our model effectively, we capture a benchmark dataset of real-world flare images with ground truth flare-free images using SONY A6100. In particular, we first capture an image with a strong light source on a tripod. We then add a lens hood to block some rays to avoid flare artifacts carefully. In addition, we also collect some flare and flare-free image pairs from the Internet to increase the diversity and the number of image pairs in the dataset. Finally, we obtain a benchmark dataset of 102 flare and flare-free image pairs.

**Baselines.** We compare our method with a traditional flare spot removal method [4], which first detects the flare

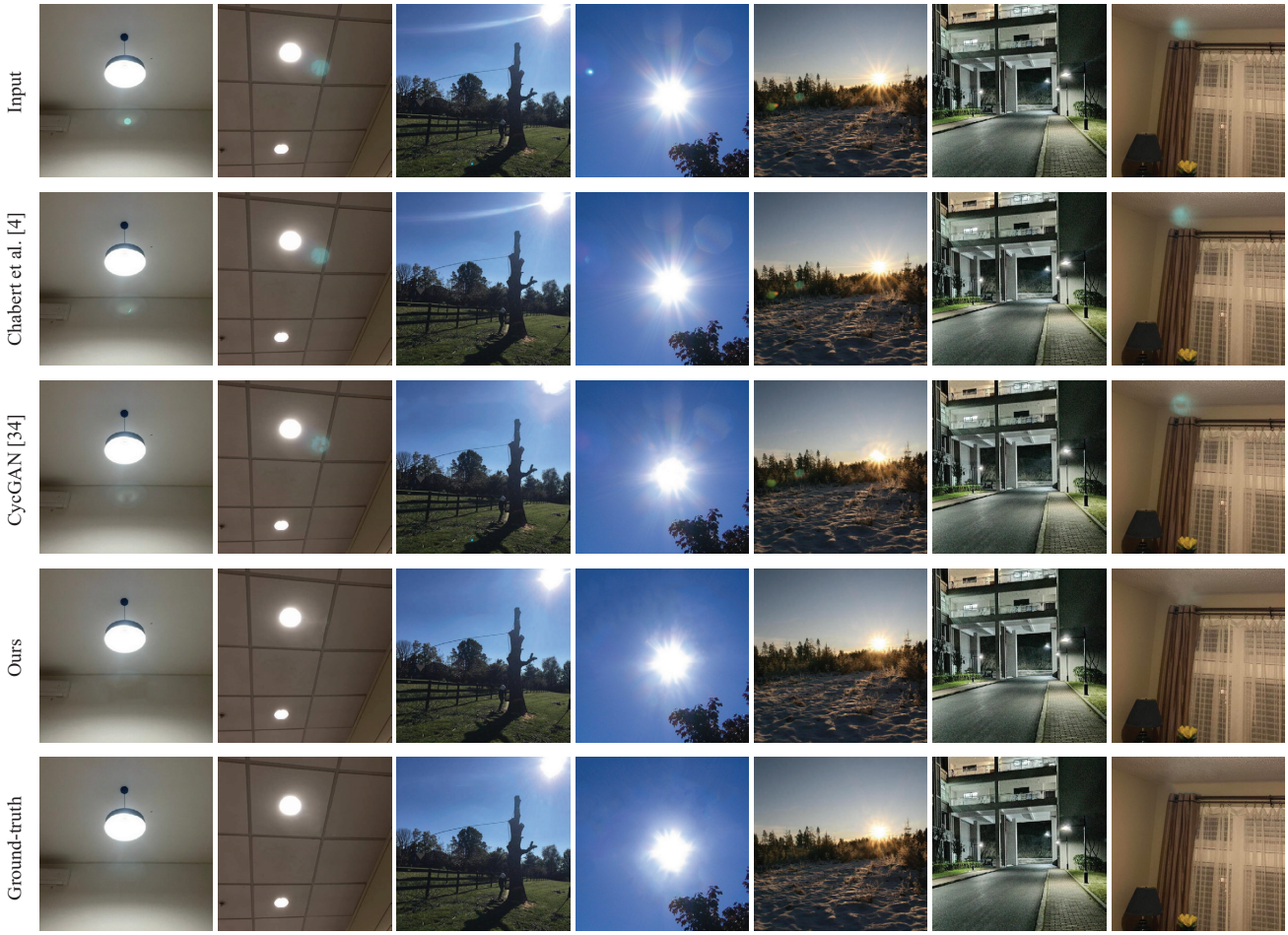



Figure 4. Qualitative comparison of the proposed method with prior works. The first and the last rows are the inputs and ground-truth flare-free images.

spot region and then removes the artifacts in an inpainting manner. We also compare our method with CycleGAN [34], which is a general image-to-image translation framework using unpaired data. We retrain the model using our training dataset so that it can map the input flare image to an output flare-free image, as in our model. To the best of our knowledge, there is only one deep learning based method [31] for SIFR, using paired training data. Unfortunately, both their paired dataset and code have not been publicly available yet. To compare with this work, we show the visual results and conduct a user study by running our model on their images.

**Evaluation Metrics.** For quantitative comparison, we compute the PSNR and SSIM metrics between the predicted flare-free images and the ground truth flare-free images to evaluate the performance of our results.

## 5.2. Comparison with Existing Methods

**Quantitative Evaluation.** We conduct a quantitative experiment on our benchmark. Table 1 compares the performance of our model with the baselines. Our method performs favorably against all baselines by a large margin on both PSNR and SSIM metrics. This suggests that our method can effectively remove flares in the images, even though it is trained only on unpaired data.

**Qualitative Evaluation.** Figure 4 shows some qualitative results of our model, compared with those from the baselines and the corresponding ground-truth. From the results, we can see that our method outperforms other methods in all cases, and removes the flare artifacts more accurately and thoroughly. Chabert *et al*. can only remove the flare spot in the image, and may sometimes fail to remove the spot artifact region completely. For example, in the first column, the flare spot artifact still exists in the image. The accuracy of the detection has a significant effect on the final removal performance. In contrast, our model can detect and remove different types of flare artifacts well.

In general, CycleGAN performs better than Chabert *et al*. due to the diverse flare artifacts learned by the network

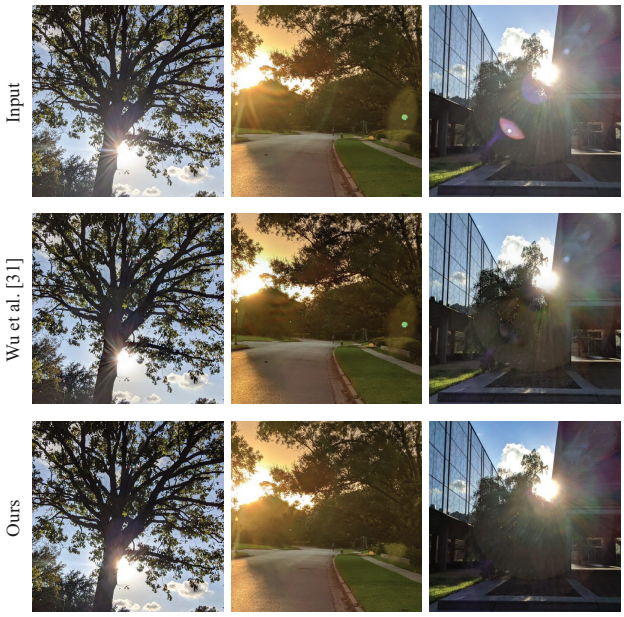



Figure 5. Visual comparison of the proposed method against the deep learning based method [31] for SIFR.

in the training process. However, it may fail to remove large flare spots in some images (*e.g.*, second and fourth columns). On the other hand, it may remove part of the light sources in other images (*e.g.*, third and sixth columns) by mis-recognizing them as flares, especially when there are multiple light sources. In contrast, due to the explicit light source guidance, our method can differentiate between light source regions and flare regions well, and can therefore remove the flares in all these examples accurately. It is worth-noting that not all flare images contain the light source (e.g., seventh column). Our method can still remove the flare artifacts even from images that do not contain any light source.

We also visually compare our model with the deep learning based method [31] for SIFR. Figure 5 shows some of the results. We can see that although our model is only trained on unpaired data, our results compare favorably with their results that are trained on paired data. For example, in the first column, we can still see unremoved flare streaks around the trunk of the tree in their result, while our method successfully removes the majority of these streaks. All these results suggest that our method trained on unpaired data works well in the SIFR task.

**User Study.** We further conduct user studies to evaluate the quality of our results. We first compare our method with Chabert *et al*. [4] and CycleGAN [34]. We randomly select 20 flare images from the benchmark dataset. We apply the above methods to these flare images to generate the corresponding flare-free images. Each time, participants are shown a flare image on the left, and three generated flare-free images by two baselines and our method on the right in random order. Participants are asked to vote for the best flare-free result. We recruit a total of 19 participants for this experiment. In the end, our results are ranked the best in 74.3% of the votes, CycleGAN in 23.6% of the votes, and Chabert *et al*. in 2.1% of the votes. This suggests that our results are preferred by the participants most of the time, confirming once again the superior performance of our model in comparison to the baselines.

We then compare our method with Wu *et al*. [30] and the baselines using the 48 images in [30]. We conduct our user study with 19 participants under the same setting as described in the above paragraph. In the end, our results are ranked the best in 43.6% of the votes, [30] in 39.5% of the votes, CycleGAN in 15.1% of the votes, and Chabert et al. in 1.8% of the votes. This further suggests that our method is more preferred.

| | PSNR↑ | SSIM↑ |
|---|---|---|
| w/o no-flare | 19.21 | 0.788 |
| w/o LSD module | 19.12 | 0.783 |
| w/o FD module | 19.88 | 0.795 |
| w/o LSD module & FD module | 18.71 | 0.778 |
| w/o light source loss | 19.37 | 0.791 |
| w/o flare loss | 20.62 | 0.801 |
| w/o adv loss | 20.19 | 0.797 |
| w/o cycle | 18.95 | 0.781 |
| Ours | **21.57** | **0.812** |

Table 2. Results of the ablation study. The best results are highlighted in bold.

### 5.3. Ablation Study

To study the effectiveness of our framework design, we create the following ablated variants for evaluation:

- w/o no-flare. We train our model without using the flare-free training images.
- w/o LSD module. We remove the LSD module from our model, *i.e*., removing the light source guidance for both flare effect removal and generation.
- w/o FD module. We remove the FD module from our model, relying only on the FR module to remove the flare artifacts.
- w/o LSD module and FD module, *i.e*., without explicit detection of light sources nor flares.
- w/o light source loss. We train the model without using the light source loss, which means the light source constraint is not enforced explicitly.
- w/o flare loss. We train the model without using the flare loss.
- w/o adv loss. We train the model without using the adversarial loss.
- w/o cycle. We train the model without using the reconstruction loss.

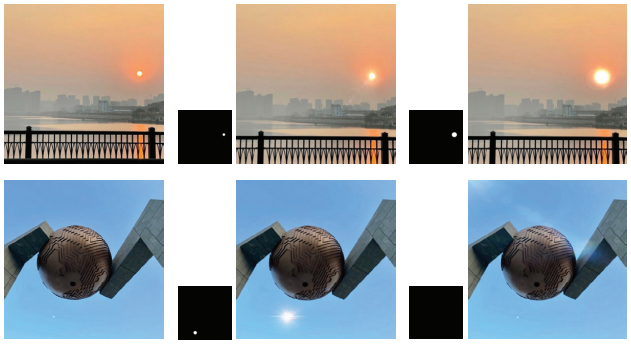



Figure 6. Flare effect manipulation. Given an input image, our method is able to generate different flare effects based on the light source mask, which indicates the shape, position, and size of the light source.

Table 2 shows the results of the ablation study. We can see that the performance drops when training the model without using flare-free images. This suggests that both flare and flare-free images can help improve the SIFR performances. We observe that without the LSD and FD modules, the network performs the worst among all the ablated models. Compared between the two ablated versions, without the LSD module has a higher performance drop than the one without the FD module. This indicates that utilizing the light source aware guidance can greatly improve the SIFR performances. Nevertheless, the performance also drops due to the removal of the FD module, confirming the advantage of explicitly detecting the flare region first before flare removal. Further, when training the network without using the light source loss, the flare loss, the adversarial loss, or the reconstruction loss, the performance would drop. This indicates that it is beneficial to incorporate additional guidance for the SIFR task.

### 5.4. Flare Effect Manipulation

We further demonstrate the applicability of our model to flare effect manipulation. Lens flares can show up in images in a variety of forms, including blobs, streaks, or colored light anywhere over the image. Understanding lens flares can help us manipulate the image in different ways, *i.e.*, adding or removing flare effects in the image. Although flare artifacts are undesirable in many images, some film directors or photographers deliberately use flares as a special effect [14] to enhance the artistic meaning of an image [11]. Flare-like effects can also help increase the perceived realism by indicating the presence of very bright light sources.

Figure 6 demonstrates this application. Given an input flare-free image, we use the FG module in our network to generate a flare image. As described before, the input to the FG module is an image and a light source mask. In this application, the light source mask can be obtained from either the input image via the LSD module (*i.e.*, automatic mode) or provided by the user (*i.e.*, interactive mode). The first row of Figure 6 shows an input image of a sunset scene, with the sun being the light source. The flare image in the middle is generated using the light source mask (the thumbnail on its left) detected automatically by the FD module, while the flare image on the right is generated using a different mask with a larger sun (the thumbnail on its left) provided by the user. Our method is able to generate different and plausible flares in both cases. The second row of Figure 6 shows an input image without an obvious light source. By using a user-provided light source mask with a sun in it, as shown in the middle (the thumbnail), our method can generate a plausible flare image. We also try using a blank light source mask, as shown on the right (the thumbnail). Our model can still generate a reasonable flare image.

From this experiment, we demonstrate that the FG module trained through our framework can be used to add flares to an image, and the appearance of the synthesized flares can be manipulated by adjusting the light source mask.

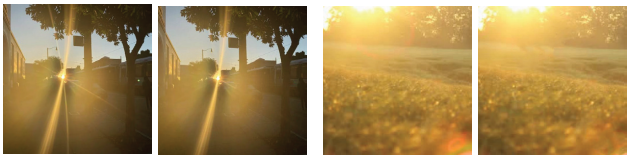



Figure 7. Failure cases. Our model may fail on some images with extremely strong light source and the flare artifacts appear all over the image.

## 6. Conclusion

In this paper, we make the first attempt to utilize light source guidance to address the SIFR problem. Our key idea is to consider the underlying relationship between the light source region and the flare region, which is learned from unpaired data with new cycle-consistency constraints. In addition, we construct the first unpaired flare and flare-free image dataset, covering diverse scenarios. Extensive qualitative and quantitative results show that our approach achieves superior performances over the baseline methods. Finally, we also demonstrate that our model can be applied to flare effect manipulation.

Although our method works well in different types of flare scenarios, it may fail on some challenging scenes where the intensity of the light source is extremely strong and the flare artifacts are all over the image, as shown in Figure 7. In this case, it is difficult for our model to distinguish between flare artifacts and the light source. A possible solution to this problem is to change the flare representation from a binary mask to an alpha matte to incorporate proper visual information. As a future work, we would like to study more types of extreme flare artifacts.

## References


[1] Understanding lens flare. https://photographylife.com/what-is-ghosting-and-flare, 2020. 1

[2] C.S. Asha, Sooraj Kumar Bhat, Deepa Nayak, and Chaithra Bhat. Auto removal of bright spot from images captured against flashing light source. In *DISCOVER*, 2019. 2

[3] Paul A. Boynton and Edward F. Kelley. Liquid-filled camera for the measurement of high-contrast images. In *Cockpit Displays X*. International Society for Optics and Photonics, 2003. 2

[4] Floris Chabert. Automated lens flare removal. 2015. 1, 2, 5, 7

[5] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE TIP*, 2004. 2

[6] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson WH Lau. Location-aware single image reflection removal. In *ICCV*, 2021. 2

[7] K. Faulkner, C.J. Kotre, and M. Louka. Veiling glare deconvolution of images produced by x-ray image intensifiers. In *International Conference on Image Processing and its Applications*, 1989. 2

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 5

[9] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, Jing Qin, and Pheng-Ann Heng. Direction-aware spatial context features for shadow detection and removal. *IEEE TPAMI*, 2019. 2

[10] Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, and Pheng-Ann Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *ICCV*, 2019. 2

[11] Matthias Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. Physically-based real-time lens flare rendering. In *ACM TOG*. 2011. 8

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 4

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[14] John Knoll. Knoll light factory. Red Giant Software. 8

[15] Hideki Kondo, Yasuhiro Chiba, and Takashi Yoshida. Veiling glare in photographic systems. *Optical Engineering*, 1982. 2

[16] Fima Koreban and Yoav Y Schechner. Geometry by deflaring. In *ICCP*, pages 1–8, 2009. 2

[17] Chao Li, Yixiao Yang, Kun He, Stephen Lin, and John E Hopcroft. Single image reflection removal through cascaded refinement. In *CVPR*, 2020. 2

[18] H. Angus Macleod. *Thin-film optical filters*. 2017. 2

[19] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 4

[20] Andreas Nussberger, Helmut Grabner, and Luc Van Gool. Robust aerial object tracking from an airborne platform. *IEEE Aerospace and Electronic Systems Magazine*, 2016. 1

[21] Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson WH Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *CVPR*, 2017. 2

[22] Ramesh Raskar, Amit Agrawal, Cyrus Wilson, and Ashok Veeraraghavan. Glare aware photography: 4d ray sampling for reducing glare effects of camera lenses. *ACM TOG*, 2008. 1, 2

[23] Sidney Ray. *Applied photographic optics*. Routledge, 2002. 1

[24] Dikpal Reddy and Ashok Veeraraghavan. Lens flare and lens glare. *Computer Vision: A Reference Guide*, 2014. 1

[25] J. Anthony Seibert, O. Nalcioglu, and W. Roeck. Removal of image intensifier veiling glare by mathematical deconvolution techniques. *Medical physics*, 1985. 2

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 5

[27] Eino-Ville Talvala, Andrew Adams, Mark Horowitz, and Marc Levoy. Veiling glare in high dynamic range imaging. *ACM TOG*, 2007. 2

[28] Patricia Vitoria and Coloma Ballester. Automatic flare spot artifact detection and removal in photographs. *Journal of Mathematical Imaging and Vision*, 2019. 2

[29] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *CVPR*, 2019. 2

[30] Tai-Pang Wu and Chi-Keung Tang. A bayesian approach for shadow extraction from a single image. In *ICCV*, 2005. 1, 2, 7

[31] Yicheng Wu, Qiurui He, Tianfan Xue, Rahul Garg, Jiawen Chen, Ashok Veeraraghavan, and Jonathan Barron. Single-image lens flare removal. *arXiv 2011.12485*, 2020. 1, 2, 4, 6, 7

[32] Zheng Zhang, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Single image veiling glare removal. *Journal of Modern Optics*, 2018. 2

[33] Hongyuan Zhu, Xi Peng, Joey Tianyi Zhou, Songfan Yang, Vijay Chanderasekh, Liyuan Li, and Joo-Hwee Lim. Singe image rain removal with unpaired information: A differentiable programming perspective. In *AAAI*, 2019. 2

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 5, 6, 7