Supplementary Information

Teacher-student learning of generative adversarial network-guided diffractive neural networks for visual tracking and imaging

Hang Su[1,2,†], Yanping He[1,2,†], Baoli Li[1,2], Haitao Luan[1,2], Min Gu[1,2], and Xinyuan Fang[1,2,*]

[1] School of Artificial Intelligence Science and Technology, University of Shanghai for Science and Technology; Shanghai, 200093, China.

[2] Institute of Photonic Chips, University of Shanghai for Science and Technology; Shanghai, 200093, China.

[†]These authors contributed equally: Hang Su, Yanping He.

Email: xinyuan.fang@usst.edu.cn

Supplementary Note 1. The principle of event-based camera and the conversion from event data to image datasets

Conventional cameras produce a series of frames at a fixed frequency. However, a significant issue arises due to the loss of crucial information between two neighboring frames. Additionally, conventional cameras necessitate substantial memory, energy expenditure, and latency, which collectively contribute to the suboptimal real-time performance of numerous algorithms.

Event cameras do not adhere to the conventional concept of "frames". In response to a change in a real scene, the Event cameras generate pixel-level outputs, or events. An event consists of $\varepsilon(x, y, t, p)$, where x, y are the pixel coordinates of the event in 2D space, t is the timestamp of the event, and p is the polarity of the event. The polarity, which can be either positive or negative, indicates the direction of the brightness change in the scene.

The operational principle of an event camera is based on detecting changes in the logarithmic light intensity at each pixel. When the change in brightness at a given pixel exceeds a certain threshold C, an "on" event is triggered if the intensity is increasing, and an "off" event is triggered if it is decreasing. This process can be mathematically expressed as:

$$\log I\left(x, y, t\right) - \log I(x, y, t - \Delta t) = \pm C \tag{S1.1}$$

where I denotes the brightness of the pixel and C is the output threshold. It is noteworthy that the threshold C is adjustable and correlates with the light sensitivity of the event camera. As a result, when recording with an event camera, a series of events $\varepsilon_i(x_i, y_i, t_i, p_i)$ with varying durations can be captured.

Since the optical diffraction neural network cannot directly process the event-based video information from event camera, it is necessary to convert this data into a two-dimensional grayscale image through dimensionality compression. In our work, we encode the polarity information of the event vectors according to Equation (S1.2), then the increase and decrease of pixel brightness are represented as two different grey levels on the greyscale image:

$$\varepsilon_i\left(x_i, y_i, t_i, p_i\right) = \begin{cases} 255, p_i = +1 \\ 0, p_i = -1 \ or \ p_i = 0 \end{cases} \tag{S1.2}$$

By stacking all the events over a period of time, we obtain an image that contains information about the motion of the target object over that period of time, which can be represented as an image:

$$E_i\left(x_i, y_i\right) = \sum_{t_i} \varepsilon_i\left(x_i, y_i, t_i\right) \tag{S1.3}$$

The obtained image E_i is denoised using a noise filter and then directly fed into the GAN-guided DNN for model training or utilized as image information for the

assessment of the trained model's efficacy.

Supplementary Note 2. Performance of GAN-guided DNN in visual tracking and imaging for the more complex scene

As shown in Figure S1a, our GAN-guided DNN exhibits high accuracy in tracking and imaging the target, even in a complex scene where the interfering car shares the same speed, profile, and comparable size as the target. Notably, the target car follows a more intricate motion trajectory, covering the entire plane. Figure S1b displays the phase distribution of the two diffractive layers in the GAN-guided DNN at the end of training. Furthermore, Figure S1c presents a quantitative analysis of the imaging performance based on structural similarity (SSIM) and peak signal-to-noise ratio (PSNR). The results indicate that the average SSIM is 0.9 and the PSNR is 21 dB, confirming that the trained DNNs maintain high accuracy for target tracking and imaging in challenging scenarios.

Supplementary Note 3. Evaluation of GAN-guided DNN performance under different lighting conditions

As shown in Figure S2, the total light power in the experiment was set to 0.5μW. We tested the model using 10%, 30%, 50%, and 100% of this total power for illumination and dataset acquisition. The first row of Figure S2c shows the event camera captures of the same dynamic action under these different lighting conditions. These recorded events are converted into formats suitable for the GAN-guided DNN training process. The output of the trained GAN and DNN for the training set data is also displayed in Figure S2c, while Figure S2b presents the imaging quality of the test results under the four lighting scenarios.

When the light was reduced to 50% of the original power, the high dynamic range of the event camera ensured that the recorded events did not blur, as would occur with traditional cameras. The GAN-guided DNN trained under this condition performed similarly to the 100% light scenario. At 30% illumination, the event-based GAN-guided DNN could still track the target, though the generated image showed slight distortions due to reduced input information. However, when the light was further reduced to 10% of the total power, the event camera recorded fewer events, as the pixel brightness variation decreased in this extremely low-light environment. Under these conditions, the GAN-guided DNN could not perform successful target tracking and imaging due to the substantial loss of input data. It is important to note that at 10% lighting, the power is only 0.05μW, which is nearly invisible to the human eye.

Supplementary Note. 4. Detailed descriptions of experimental data acquisition systems and GAN-guided DNN test setup

Figure S3a illustrates the optical setup for data acquisition using event-based cameras. The video is decomposed into individual frames, loaded onto a DMD, and switched at 400 fps. A 4f system is used to conjugate the DMD's imaging surface with the sensing surface of the event-based camera, generating high-quality motion video. The event-based camera captures and converts this into a data format trainable by the GAN-guided DNN. DMD: Digital Micromirror Device (DLP9000X, Texas Instruments); Event-based camera (DAVIS346, iniVation). The wavelength of the incident continuous light is 532 nm.

Figure S3b shows the optical setup for the GAN-guided DNN test system. The incident continuous wave at a wavelength of 632.8 nm is generated by a He-Ne laser (CW, HNL210L, Thorlabs) with a power output of 14.4 mW. The beam is spatially filtered and systematically irradiated to SLM1 (X13138, Hamamatsu). To load the amplitude information of the input image on the spatial light modulator (SLM), two quarter-wave plates (QWPs) with orthogonal fast axes are mounted on the front and back of the SLM1 (X13138, Hamamatsu). The next two SLMs (SLM 2 and SLM 3, X13138, Hamamatsu) with a spacing distance of 150 mm are utilized to construct diffractive neural networks for visual tracking and imaging of the interested moving target. The output image is captured by the CCD camera (acA2040-90uc, Basler).

Supplementary Note 5. The comparison between the GAN-guided DNN and a DNN without GAN-guided

Figure S4 illustrates different frames of a motion video recorded by an event camera, with the output of various events converted into images. These images serve as inputs for training both the GAN-guided DNN and the DNN without GAN guidance. As observed, the event camera captures dynamic actions but may miss certain information. Consequently, the DNN without GAN guidance, trained directly on these incomplete images, only retains the position information of the tracking target and lacks the ability to repair missing data. In contrast, the GAN-guided DNN benefits from the GAN's generator to reconstruct the missing information, providing more accurate shape details of the tracking target. We use PSNR to evaluate the tracking and imaging accuracy of both models. Figure S4b shows that, without GAN guidance, the PSNR between the standard DNN's output and the target is approximately 15 dB. In comparison, the PSNR for the GAN-guided DNN is about 22-23 dB, indicating that the GAN-guided DNN performs significantly better in imaging tasks, successfully addressing challenges that the standard DNN struggles with.

Supplementary Note 6. The quantitative analysis of GAN-guided DNN and existing visual tracking and imaging solutions

The existing solution uses the traditional frame-based camera as the data acquisition terminal, with captured images processed as input for network training. In contrast, our work utilizes an event camera for information acquisition. The key differences are summarized in Table 1.

It can be observed that using the event camera for information input offers significant advantages in terms of output frequency, energy consumption, and dynamic range.

Table 1. Frame camera and event camera

	Frame camera	Event camera
Frequency	<64	~10e6
Energy consumption	>2 W	<10 mW
Dynamic range	60 dB	140 dB

After receiving the input information, both the GAN-guided DNN and the existing method, which only uses GAN for visual tracking, rely on computer-based network training, with no significant differences between them in this process. However, during testing or application, the existing solution performs visual tracking and imaging tasks using traditional computer architecture, whereas the GAN-guided DNN leverages the diffraction effect of light. The key differences between the two approaches are presented in Table 2.

Table 2. The difference between GAN-only solution and GAN-guided DNN

	GAN-only	GAN-guided DNN
Computing speed	>10e-2 s	~10e-5 s
Energy consumption	~300 W	10~20 W

It is important to note that the current limitations on the computing speed and energy consumption of the GAN-guided DNN stem not from the model itself, but from the hardware system. For example, due to the extremely fast speed of light, the information transmission in a DNN is also incredibly fast, with almost zero delay. As a result, the image loading speed (e.g., SLM and DMD image upload speed) becomes the bottleneck for computing speed. Similarly, while the energy consumption of GAN networks is primarily due to GPU usage, the energy consumption of GAN-guided DNNs is mainly driven by the laser and DMD operation. Nonetheless, in terms of both computational speed and energy efficiency, the GAN-guided DNN far surpasses existing visual tracking and imaging solutions.

Supplementary Note 7. The construction and the training details of GAN

The training of a GAN is an unsupervised process, typically using a stochastic gradient descent algorithm. During training, random noise *z* is often sampled from a uniform or Gaussian distribution, known as the prior distribution $P_Z(z)$. This random noise serves as the input to the generator *G*, which then produces new data *G(z),* effectively mapping the noise to a new data space. This generated data *G(z)* is assumed to follow the distribution $P_g(z)$. The input to the discriminator *D* consists of two sets of data: one is the generated data *G(z)*, and the other group is the real-world data *x*, which follows the distribution $P_{data}(z)$. The role of the discriminator is to evaluate the authenticity of these two sets of data by outputting a scalar probability, representing how likely a given input belongs to the real data distribution. A higher value indicates that the generated data *G(z)* closely resembles the real data *x*. If the distributions of the real and generated data are very different, the discriminator can easily distinguish between them[62].

Thus, during training, the objective of the discriminator *D* is to maximize *D(x)*, the probability that real data is correctly identified, while minimizing *D(G(z))*, the probability that generated data is incorrectly classified as real. Meanwhile, the goal of the generator *G* is to make the distribution *G(z)* of the generated new data as close as possible to the real data distribution $P_{data}(x)$, so that the discriminator *D* cannot distinguish the newly generated data from the real data. Therefore, the objective function for optimizing the generating adversarial network can be expressed in Equation (S7.1):

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}\left[\log\left(D(x)\right)\right] + E_{z\sim P_Z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad \text{(S7.1)}$$

During training, the two networks engage in a dynamic "game process". In order to provide a sufficient learning gradient for both the generator *G* and the discriminator *D*, the natural logarithm function is usually used. For the loss function, a larger *log(D(x))* indicates a higher probability that the discriminator *D* correctly classifies real data *x* as real, meaning better performance. Conversely, a larger *D(G(z))* implies a higher likelihood that *D* incorrectly classifies the fake data generated by *G* as real, reflecting poorer performance. The term *1-D(G(z))* represents the probability that *D* correctly identifies the fake data, and thus, a larger *log(1-D(G(z)))* indicates stronger performance by the discriminator.

Therefore, when training the discriminator *D*, the objective is to maximize *log(D(x))* and *log(1-D(G(z)))*, ensuring the network can distinguish fake data from real data with the highest accuracy. In contrast, when training the generator *G*, the goal is to minimize *log(1-D(G(z)))*, even though this maximizes the loss for the discriminator. In this process, two networks need to be iteratively trained alternately, one network is fixed, and the parameters of the other network are optimized. The purpose of training is to maximize the loss of the other network. Finally, the fake data generated by *G* becomes increasingly indistinguishable from the real data, causing *D* to struggle in making a distinction.

The process begins by sampling m noise inputs from the noise distribution $P_g(z)$ alongside m real data samples from the real data distribution $P_{data}(x)$. These m samples form the minimum training batch. Next, the generator network held is fixed while the discriminator is trained using stochastic gradient ascent algorithm to update its parameters, thereby maximize *log(D(x))* and *log(1-D(G(z)))*. The corresponding expression is as follows:

$$\nabla \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^i\right) + \log\left(1 - D\left(G\left(z^i\right)\right)\right) \right] \tag{S7.2}$$

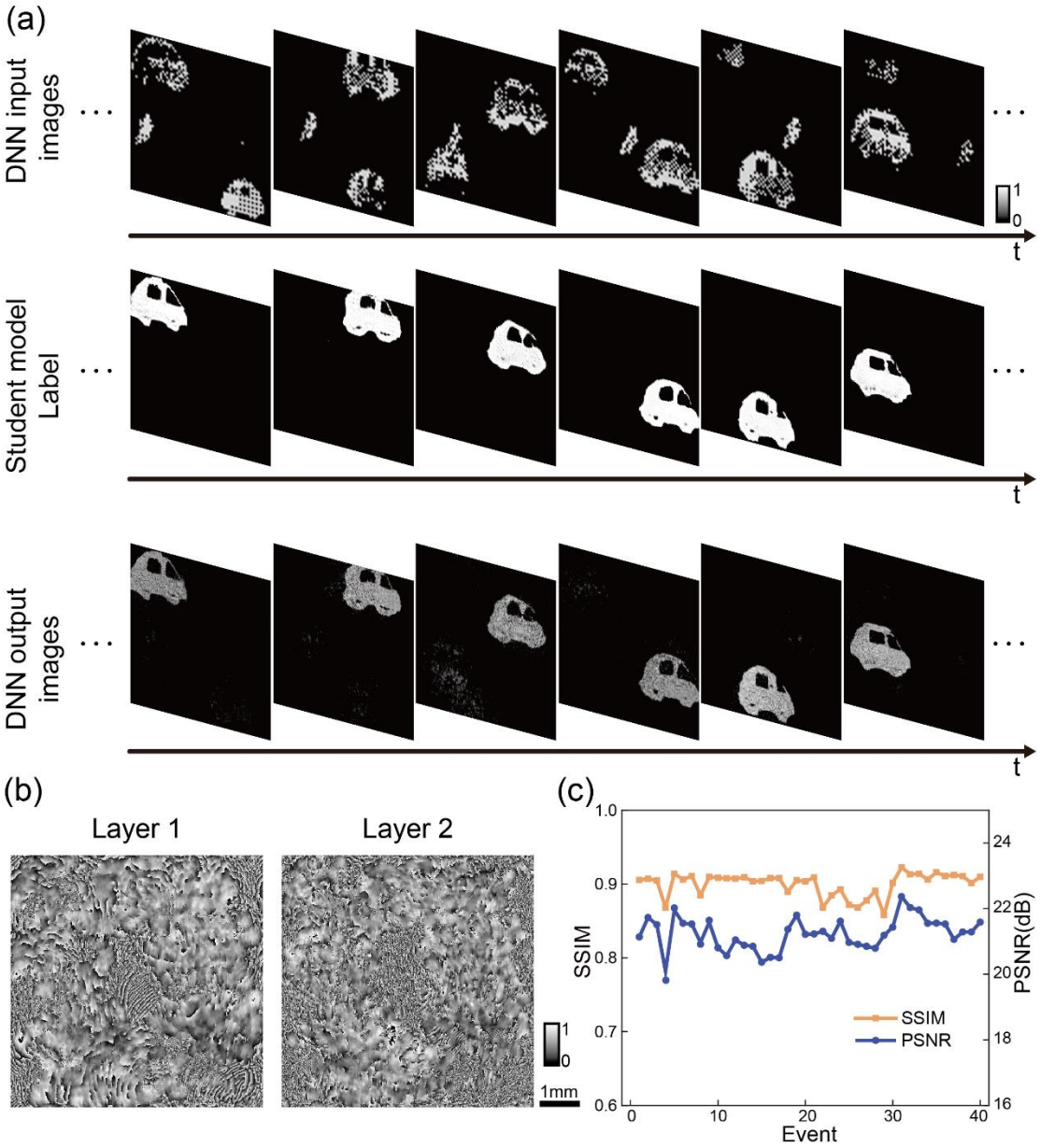
where i represents the *i-th* sample from the m randomly selected samples, and ∇ denotes the derivative.

After completing one iteration of the training process, m noise samples are randomly selected from the new noise data distribution. With the parameters of the discriminator network D fixed, and the parameters of the generator network G are updated. Then, the random gradient descent algorithm is used, that is, to minimize *log(1-D(G(z)))*. The corresponding expression is as follows:

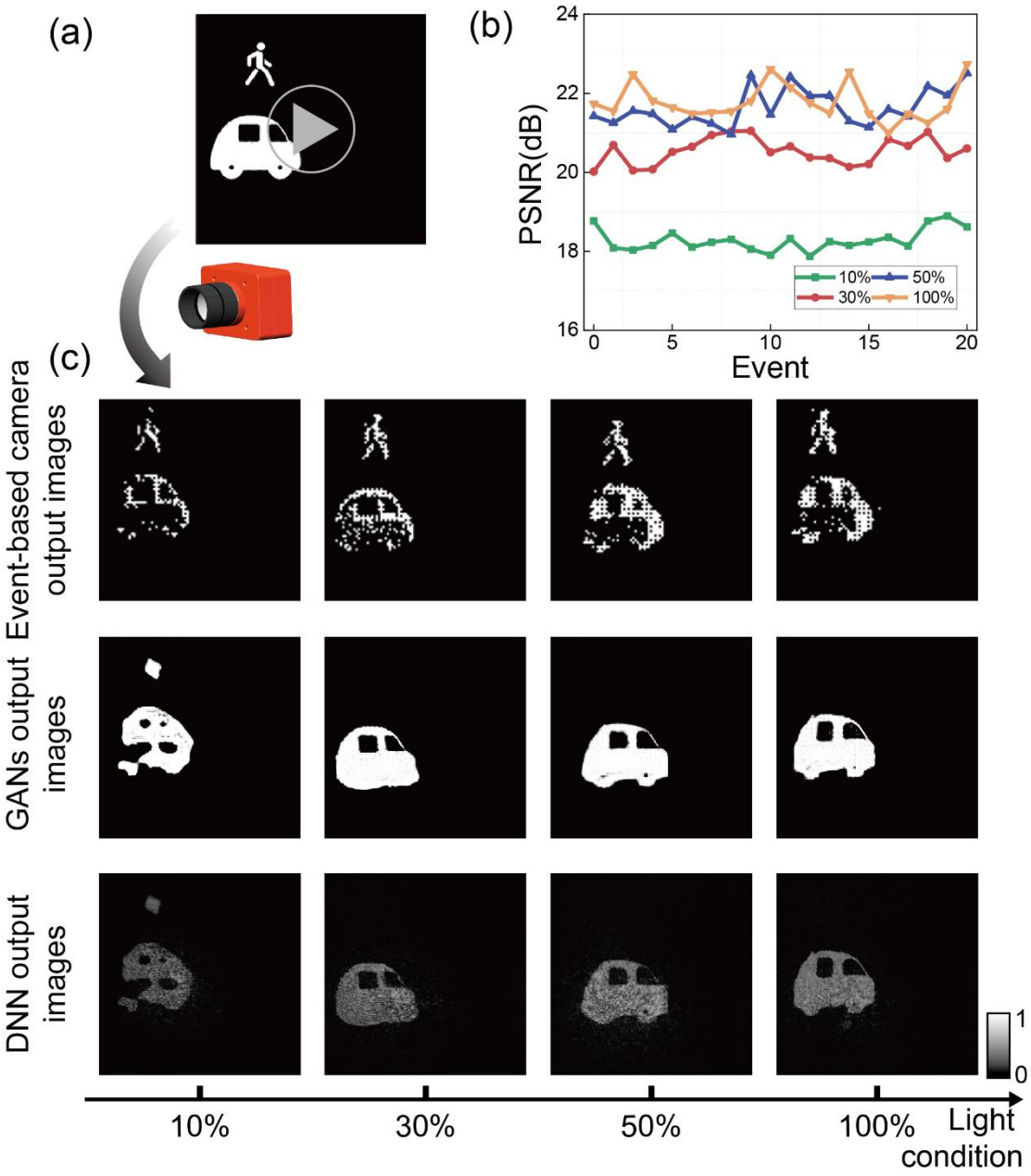
$$\nabla \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^i\right)\right)\right) \tag{S7.3}$$

The variable i in Equation (S7.3) carries the same meaning as in Equation (S7.2).

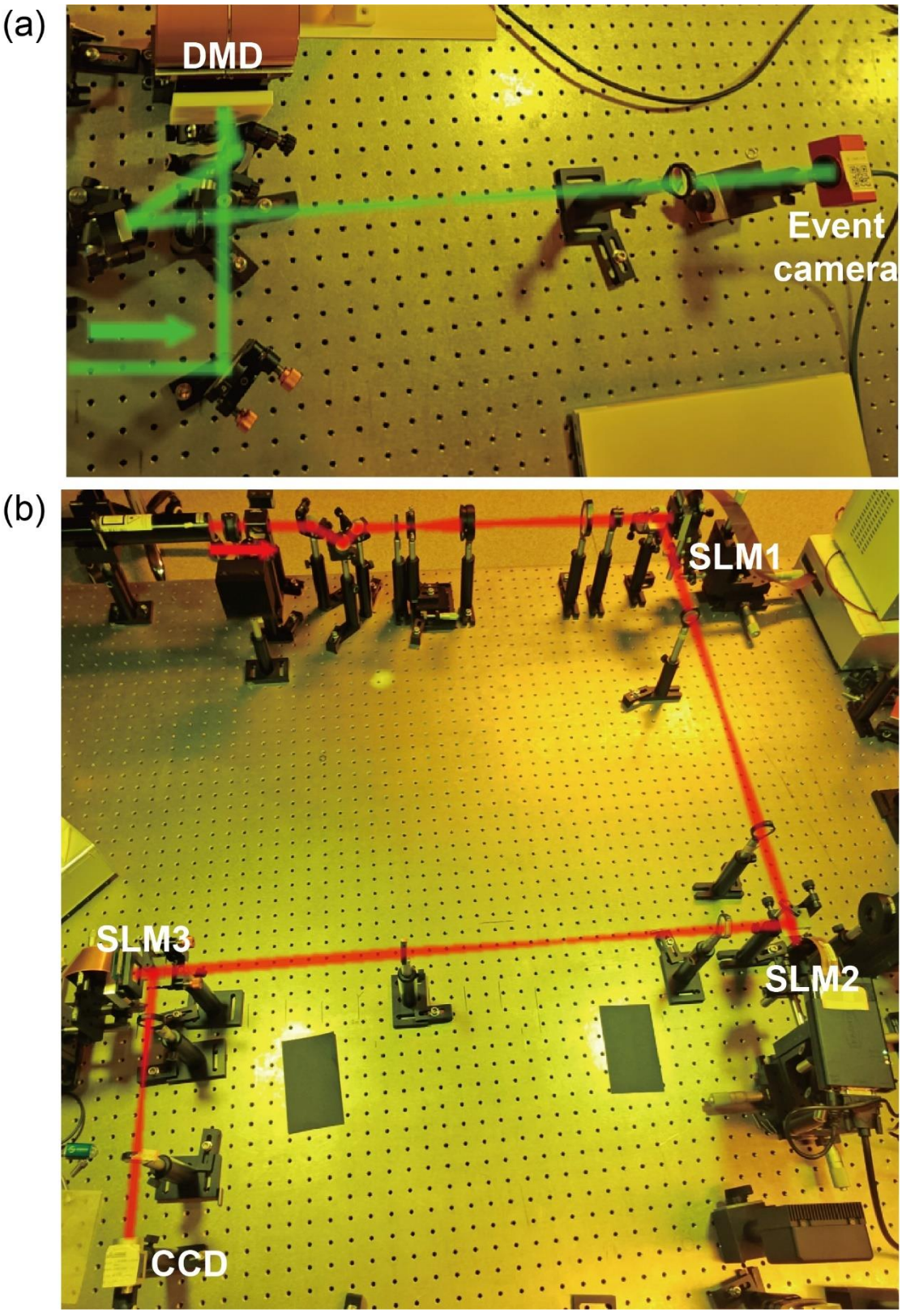
When the data distribution of the generated data, $P_g(z)$, aligns with that of the real data, $P_{data}(x)$, the network model reaches an optimal solution. In this case, the generator demonstrates a strong ability to produce data nearly indistinguishable from real data. Similarly, the discriminator exhibits high learning capacity but is unable to reliably distinguish between fake data generated by the generator and actual real data. Thus, the network successfully generates synthetic data that closely approximates the real data.



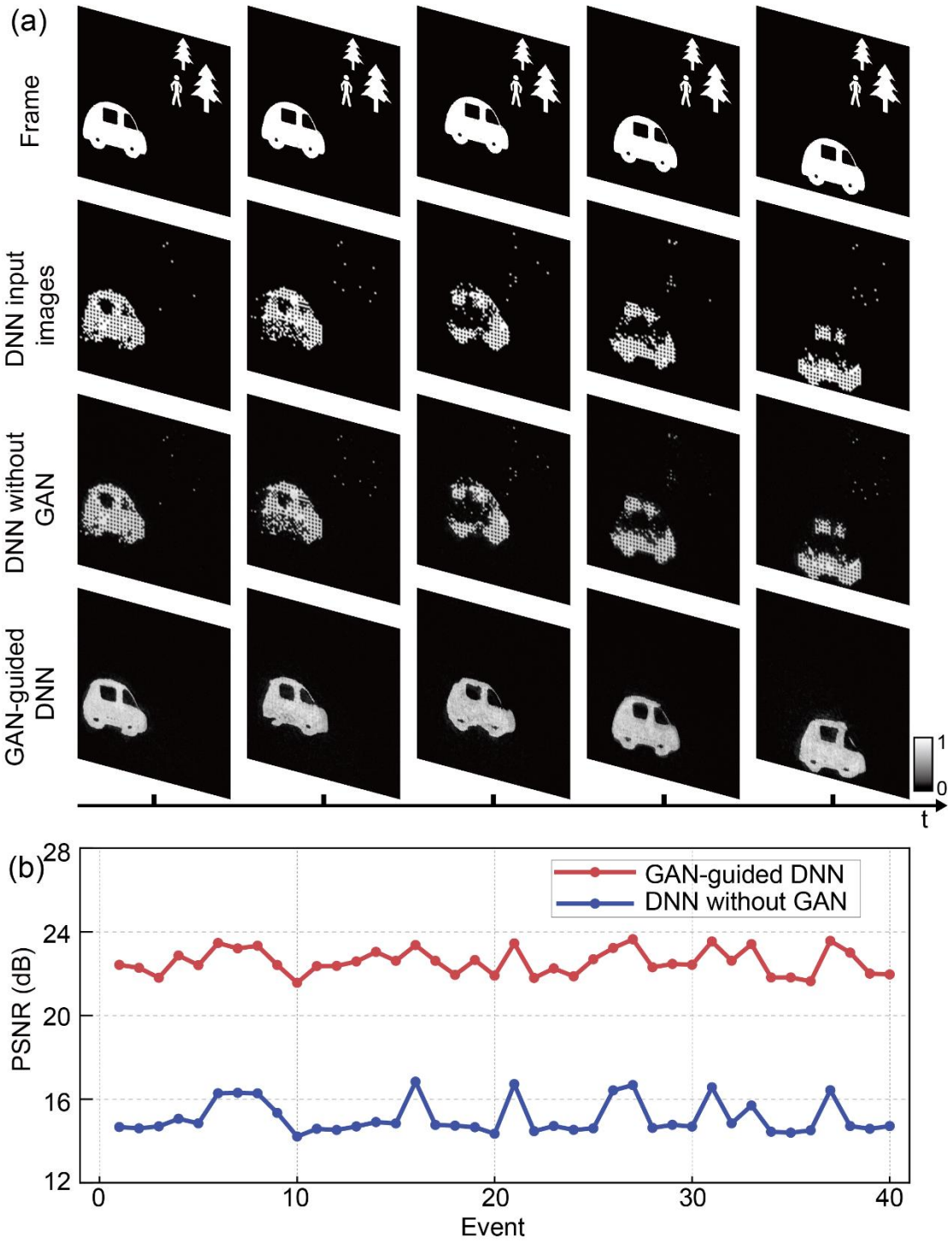
Supplementary Figure S1. Performance testing of GAN-guided DNNs for visual tracking and imaging in complex scenarios. (a) Visual tracking and imaging results of the GAN-guided DNN for target cars in a complex scene, which includes a moving pedestrian and two similarly shaped cars. (b) Phase distributions of the two diffraction layers in the double-layer DNN trained under GAN guidance. (c) SSIM and PSNR values of the GAN-guided DNN's imaging results compared to the labels in the test set for complex scenes.



Supplementary Figure S2. Visual tracking and imaging results from the GAN-guided DNN under varying illumination conditions of 10%, 30%, 50%, and 100% of the total light power (0.5μW). (a) A dynamic motion scenario. (b) PSNR values of the GAN-guided DNN outputs under different lighting conditions. (c) The training process of GAN-guided DNN across varying lighting conditions.



Supplementary Figure S3. The picture of the experimental demonstration system. (a). The optical setup and light path for data set collection. (b). The optical setup and light path for GAN-guided DNN testing.



Supplementary Figure S4. Visual tracking and imaging results for the same target using GAN-guided DNN versus DNN without GAN. (a) The training results for the visual tracking and imaging of the target car. (b) PSNR values for the same test object, comparing the imaging quality and accuracy of GAN-guided DNN and DNN without GAN.