

From Noise to Events: Conditional Diffusion for Event Data Augmentation

Ruofei Wang¹, Ziyuan Luo¹, Peiqi Duan², Xiufeng Huang¹,
Boxin Shi², and Renjie Wan^{1*}

¹ Department of Computer Science, Hong Kong Baptist University

² School of Computer Science, Peking University

{ruofei, ziyuanluo, xiufenghuang}@life.hkbu.edu.hk, {duanqi0001, shiboxin}@pku.edu.cn, renjiewan@hkbu.edu.hk

Abstract. Event vision exploration holds significant achievements in various vision tasks, such as object recognition, action recognition, *etc.* However, improving the model’s robustness against overfitting remains a critical challenge. To address this, we propose augmenting event data from the noise rather than applying fixed data transformations. Specifically, we first decouple the quadruplet event data across spatial, temporal, and polarity dimensions to construct a representation, breaking existing frame-tied manners. Then, an event diffusion decoder is integrated with a condition extractor to gradually optimize the noise to the event data (N2E). The condition extractor aims to yield guidance vectors from the original data, which are fed into each optimization step of the diffusion process. During data augmentation, we can directly generate new event streams by adjusting the noise, diffusion parameters (*e.g.*, number of steps or noise intensity), or conditioning vectors, thereby boosting the performance of downstream event vision models. Extensive experiments on two downstream tasks among five event vision models demonstrate that N2E achieves greater improvements than existing augmentation baselines. Code is available at <https://github.com/rfww/n2e>.

Keywords: Event data augmentation · Event data generation · Denoising diffusion

1 Introduction

Event cameras are bio-inspired vision sensors that capture a scene by only detecting changes in light intensity [5]. Such a special mechanism leads to event data’s sparse nature, showing great advantages in various vision tasks [13, 29, 34, 53, 55]. Despite the great achievements in these fields, studying event-based data augmentation methods [12, 38, 43] to address the potential overfitting is underexplored compared to frame-based methods.

² Peiqi Duan and Boxin Shi are with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University. *Corresponding author.

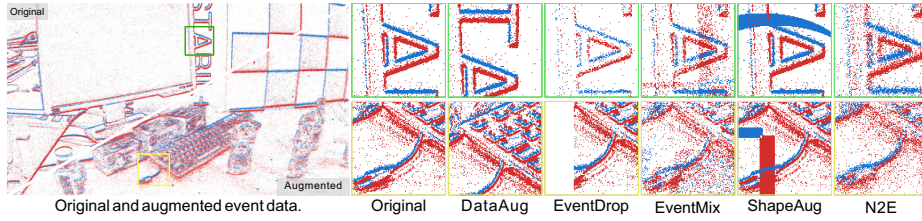


Fig. 1: Visual examples and comparisons when augmenting a 1K-resolution event data [24]. Our N2E generates controllable motion blur and background noise of varying intensity. In contrast, existing methods either provide only limited diversity (DataAug [9]: coordinate translation), destroy temporal continuity (EventDrop [12]: random drop by area or time bins), introduce unbalanced label assignment (EventMix [38]: mixing random events), or add unrealistic artifacts (ShapeAug [4]: simulated event injection). Blue and red denote positive and negative event, respectively.

Inspired by image augmentation techniques [49, 50], several studies have introduced event-based version [6, 12, 38, 43] to tackle this challenge. For example, EventDrop [12] randomly discards a portion of events to enhance noise robustness. EventMix [38] blends two event streams with a mask in spatiotemporal space to render new data. However, these methods show sub-optimal performance due to the corrupted temporal continuity (EventDrop [12]) and ambiguous label assignment (EventMix [38]), as shown in Fig. 1. These issues stem from the fundamentally different data structure of event data compared with standard images.

As shown in Fig. 1, an event is depicted as a quadruplet (x, y, t, p) , where (x, y) denote the spatial coordinates, t indicates the time stamp, and p represents the polarity. The polarity p shows a binary nature (± 1), which indicates whether the brightness increases or decreases. Due to this unique data structure, directly masking a spatial region for augmentation inevitably affects other dimensions, thereby easily breaking temporal continuity, which is of great significance for some event representation approaches [9, 25]. Mixing-based solutions address it by employing a 3D mask to guide the merging of two events in both the spatial and temporal dimensions. However, setting the new label with the mask size and event count leads to imbalanced distributions for the data with slow motions (fewer events).

Therefore, a more feasible way is to generate new event data by considering its four dimensions simultaneously, thereby maintaining spatiotemporal continuity. Importantly, the injected noise should remain label-independent to prevent potentially unreasonable label assignments. To achieve this, we propose a generation strategy to diversify the event data volume by leveraging the conditional Denoising Diffusion Implicit Model (DDIM) [42]. Such a model employs a bidirectional noise optimization process to synthesize the target data, naturally aligning with the concept of data augmentation [40]. Users can easily control parameters of the diffusion model to generate different data, thus effectively enriching the event dataset. In this process, two key factors should be considered:

1) The augmentation should be applied directly to the raw event data instead of intermediate representations, ensuring flexibility in adopting various event representation methods for downstream tasks. 2) The generated data should follow the same distribution as the original samples to prevent the introduction of negative examples that could impair downstream models.

In this paper, we propose a new event data augmentation method that converts noise to event data, named **N2E**. Firstly, we partition the event stream into slices containing Δ_n consecutive events and represent a slice as a 4-channel 3D tensor. This representation preserves all event dimensions concurrently, thereby maintaining the temporal continuity. Subsequently, a condition extractor is employed to extract key messages from original events to guide a conditional DDIM to denoise. Owing to the intrinsic randomness among events, directly using random noise sampled from the Gaussian distribution to synthesize data during the inference phase is challenging. Therefore, we learn from [35] to reuse the diffusion model to estimate a random noise from the original event data as the input to generate new samples. As for event data augmentation, we can easily regulate the data diversity by tuning parameters of the N2E, such as the number of diffusion steps, noise scalability, or conditional messages. N2E allows targeted augmentation of those overfitting-prone samples, thereby improving the augmentation efficiency. Our main contributions are as follows:

- We introduce N2E, a novel event data augmentation framework that augments event data from the noise, effectively increasing data volume.
- We present a new event representation approach, preserving all four dimensions simultaneously without potential event loss.
- Extensive experiments show that our N2E achieves competitive performance in event-based object classification and action recognition.

2 Related work

2.1 Event vision

Event vision has emerged as a promising paradigm in computer vision, leveraging neuromorphic sensors that asynchronously capture pixel-level brightness changes rather than conventional frame-based imaging. This event-driven approach offers several distinctive advantages, including exceptional temporal resolution, high dynamic range, minimal latency, and reduced power consumption [37, 47, 57]. The growing availability of diverse event-based datasets [1, 10, 21, 26, 34, 45] has catalyzed research across multiple vision tasks. To bridge the gap between asynchronous event streams and existing deep learning architectures, various representation methods [9, 16, 46] have been developed to convert sparse event data into dense formats compatible with standard neural networks. Recent work has demonstrated the superiority of event cameras in challenging scenarios where traditional RGB sensors struggle, particularly in low-light conditions and high-speed motion capture [33]. Building upon these foundations, the field has witnessed substantial progress across diverse applications, including semantic seg-

mentation [44], pose estimation [18], image restoration tasks such as deblurring [19, 23, 52] and denoising [8], optical flow estimation [30, 39, 56], and object recognition [22, 57]. Yet, realizing the full potential of event-based vision models hinges on mitigating overfitting arising from data scarcity, underscoring the need for effective event data augmentation.

2.2 Event data augmentation

The augmentation of event data presents unique challenges due to its asynchronous, sparse nature and inherent spatiotemporal structure. Early efforts like EventDrop [12] introduced stochastic event removal to enhance model generalization, while NDA [28] systematically investigated the applicability of conventional image augmentation techniques to event streams. More recent works have explored region-based strategies, including EventMix [38] which performs Gaussian-sampled region replacement between samples, and EventRPG [43] which leverages class activation mapping and saliency detection to guide region cropping. Other notable approaches include ShapeAug [4], which augments data by translating geometric primitives across the sensor plane, EventAugment [11], which automatically searches for optimal augmentation combinations, and EventZoom [6], which introduces temporal progressive scaling and shifting to preserve spatiotemporal coherence. Despite these advances, existing methods predominantly operate on individual event attributes in isolation, manipulating spatial coordinates, time stamps, or polarities independently. This overlooks the fundamental interdependence among these dimensions and potentially disrupts the intrinsic spatiotemporal continuity that characterizes event data. Thus, we propose a dimension-decoupled representation that preserves such an interdependence and is coupled with a generation model for event augmentation.

2.3 Diffusion models

Diffusion models [15, 42] have demonstrated outstanding performance in various fields. Such models sample stochastic noise from a Gaussian distribution and recover the target data by a series of iterative denoising steps. In detail, the diffusion process (forward) gradually adds noise to data, while eliminating the noise by a sequence of transitional steps during the reverse process. The entire process in transitional steps ($\tau \in [0, N]$) can be formulated as:

$$\begin{aligned} q(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}) &= \mathcal{N}(\mathbf{x}_\tau; \sqrt{1 - \beta_\tau} \mathbf{x}_{\tau-1}, \beta_\tau \mathbf{I}); \\ p_\theta(\mathbf{x}_{\tau-1} | \mathbf{x}_\tau) &= \mathcal{N}(\mathbf{x}_{\tau-1}; \mu_\theta(\mathbf{x}_\tau, \tau), \sigma_\tau^2 \mathbf{I}). \end{aligned} \quad (1)$$

β_τ denotes the variance schedule, a positive constant that governs the noise magnitude in the diffusion process. The reverse (denoising) process aims to predict the posterior mean $\mu_\theta(\mathbf{x}_\tau, \tau)$ by a learned neural network. Generally, σ_τ^2 is a time-dependent constant. In Denoising Diffusion Implicit Model (DDIM) [42], the denoised data $\mathbf{x}_{\tau-1}$ can be computed from the noisy data $\mathbf{x}_\tau$ with:

$$\mathbf{x}_{\tau-1} = \sqrt{\alpha_{\tau-1}} \left(\frac{\mathbf{x}_\tau - \sqrt{1 - \alpha_\tau} \epsilon_\theta^\tau}{\sqrt{\alpha_\tau}} \right) + \sqrt{1 - \alpha_{\tau-1} - \sigma_\tau^2 \epsilon_\theta^\tau + \sigma_\tau \epsilon_\tau}, \quad (2)$$

where ϵ_τ is a forward pass of the diffusion model, which takes $\mathbf{x}_\tau$ and the current timestep τ as input, and $\tilde{\mathbf{x}}_0 = f(\epsilon_\tau(\mathbf{x}_\tau, \tau), \tau)$ is an estimation of the fully denoised data which is computed from the output of the diffusion model and the current timestep. These generative capabilities have been successfully applied to image synthesis [2, 3, 36], video generation [17], 3D content creation [54], image restoration [27, 48], and image editing [20]. However, absorbing insights from such probabilistic models to generate asynchronous event data is challenging.

3 Methodology

3.1 Preliminary

Event cameras asynchronously detect per-pixel logarithmic brightness changes with microsecond-level resolution to produce sparse event data, which is formally defined as:

$$\mathbf{E} = \{\mathbf{e}_k\}_{k=1}^N = \{(x_k, y_k, t_k, p_k)\}_{k=1}^N, \quad (3)$$

where (x_k, y_k) are pixel coordinates, t_k is the time stamp, and $p_k \in \{+1, -1\}$ is the polarity (ON/OFF). An event $\mathbf{e}_k$ fires when:

$$|\log L(x_k, y_k, t_k) - \log L(x_k, y_k, t_k - \Delta_t)| > \sigma, \quad (4)$$

with $p_k = +1$ if the change is positive, else -1 . Δ_t denotes a temporal bin. The raw event data is stored as a highly sparse quadruplet with: coordinates (x, y) , time stamp (t) , and polarity (p) . This unique characteristic leads to key benefits include ultra-low time latency and high efficiency. To be compatible with standard deep vision models, the quadruplet is transformed into dense frame-like representations, such as Event Frame [31], Time Surface [25], Event Spike Tensor [9], *etc.* The representation can be formulated as:

$$\mathbf{R}(x, y, c) = \sum_{k=1}^N \cdot \kappa_s(x - x_k, y - y_k) \cdot \kappa_t(t_k; t_c) \cdot f(p_k), \quad (5)$$

where $\mathbf{R} \in \mathbb{R}^{H \times W \times C}$ denote the output representation, $\kappa_s(\cdot)$ and $\kappa_t(\cdot)$ represent the spatial and temporal kernel functions respectively, $f(\cdot)$ indicates the polarity mapping function. H and W are the height and width of the representation. $c \in \{1, \dots, C\}$ is the index of the output channel. This representation is a lossy transformation, where events sharing the same coordinates will be merged or discarded. Such a compromise may not impair downstream tasks like object recognition. However, in event data augmentation, it degrades dense temporal details, thereby widening the domain gap between original and generated data. Therefore, employing existing representations for event generation is infeasible.

3.2 Event generation from noise

Overview. The overall pipeline of our N2E is shown in Fig. 2. The framework takes the original event stream as input and outputs a new generated one.

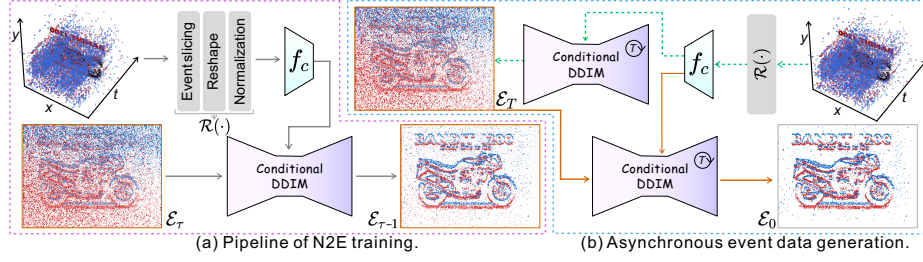


Fig. 2: Overview of N2E. (a) During training phase: Original event data undergoes event slicing, reshaping, and normalization to produce condition c . Then, such a condition is to guide the denoising process for learning a event-to-event diffusion model. (b) During inference: We sample Gaussian noise $\mathcal{E}_T$ using the diffusion model’s reverse process and combine it with the condition c to generate augmented event data. Various event data can be generated with different diffusion parameters and condition vectors.

Specifically, we implement a conditional DDIM event decoder $p(\mathcal{E}_{T-1} | \mathcal{E}_T, \mathbf{c})$ that learns the reverse diffusion process over the event tensor. In practice, this means the decoder progressively denoises the intermediate event variable $\mathcal{E}_T$ to produce $\mathcal{E}_{T-1}$ at each step, guided by the condition $\mathbf{c}$. Here $\mathbf{c}$ denotes the condition information extracted from the input event stream $\mathcal{E}_0$. This conditional input is designed to capture the key context of the original event sequence and use it to steer the noise estimation. Such conditioning not only helps the decoder generate event streams that are consistent with the input’s semantics, but also provides a direct handle for data augmentation: by modifying the condition $\mathbf{c}$, we can produce diverse, yet contextually valid, variations of the event stream. To generate new event data, first, we convert the asynchronous event data into an event stack; second, we feed this stack into the DDIM decoder to estimate the noise $\mathcal{E}_T$; third, integrating such stochastic noise with the condition $\mathbf{c}$ renders new event data with different number of diffusion steps, conditions, *etc.*

Event representation. To enable the event data be compatible with diffusion models, we present a dimension-decoupling transformation strategy. We first perform slicing along the four dimensions using a predefined event length Δ_n to extract the sub-event. The resulting 4D sub-event is partitioned according to each dimension and then reshaped into a compact 2D tensor with the shape of $h \times w$ in each dimension as:

$$\mathcal{E} = \mathcal{R}(\mathbf{E}, \Delta_n), \quad (6)$$

where $\mathcal{E} \in \mathbb{R}^{n \times 4 \times h \times w}$ in subject to $n = \lfloor N/\Delta_n \rfloor$. $\mathcal{R}(\cdot)$ denotes the representation function with channel normalization. This transformation effectively represents a series of quadruplets into an image-like representation that retains both sparsity and precise temporal dynamics. Meanwhile, the transformation only mentions event slicing, reshaping, and normalization, thereby ensuring efficiency. After representing the event into a regular pattern, we employ a probabilistic model to optimize the inherent noise contained in the event data.

Event decoder is a conditional DDIM that models the reverse-diffusion step as a Gaussian whose mean is predicted by a neural network conditioned on c . Concretely, for each timestep τ we define:

$$p_\theta(\mathcal{E}_{\tau-1} \mid \mathcal{E}_\tau, c) = \mathcal{N}(\mathcal{E}_{\tau-1}; \mu_\theta(\mathcal{E}_\tau, c, \tau), \beta_\tau \mathbf{I}), \quad (7)$$

where $\beta_\tau = 1 - \alpha_\tau$ is the variance at step τ , and the mean μ_θ is given by the usual DDIM reparameterization. In particular, letting $\bar{\alpha}_\tau = \prod_{i=1}^\tau \alpha_i$, we set:

$$\mu_\theta(\mathcal{E}_\tau, c, \tau) = \frac{1}{\sqrt{\alpha_\tau}} \left(\mathcal{E}_\tau - \frac{\beta_\tau}{\sqrt{1 - \bar{\alpha}_\tau}} \epsilon_\theta(\mathcal{E}_\tau, c, \tau) \right). \quad (8)$$

The $\epsilon_\theta(\mathcal{E}, c, \tau)$ denotes the noise predicted by the neural network (a UNet), which is explicitly conditioned on c . To extract condition c , a deep condition extractor ($f_c(\cdot)$) is implemented with residual convolution blocks and attention modules, where $c = f_c(\mathcal{E})$. The condition c is designed to steer the event decoder to denoise, thereby providing a key for data personalization.

For training our event decoder and condition extractor, the loss function is:

$$\mathcal{L} = \sum_{\tau=1}^T \mathbb{E}_{\mathcal{E}_0, \epsilon_\tau} \left[\lambda_1 \|\epsilon_\theta(\cdot) - \epsilon_\tau\|_2 + \lambda_2 \|\epsilon_\theta(\cdot) - \epsilon_\tau\|_1 \right], \quad (9)$$

where $\epsilon_t \in \mathbb{R}^{4 \times h \times w} \sim \mathcal{N}(0, \mathbf{I})$, $\mathcal{E}_\tau = \sqrt{\alpha_\tau} \mathcal{E}_0 + \sqrt{1 - \alpha_\tau} \epsilon_\tau$, T denotes the number of diffusion steps, and λ indicates the balanced parameter.

Event data augmentation. Generally, the trained DDIM has great generation ability to render diverse patterns from random Gaussian noise. However, for event data, it shows a very random characteristic. Thus, the trained decoder is hard to directly generate a satisfactory event stream from the random Gaussian noise. To this end, we learn from [35] that employs the DDIM to transfer an event data into a stochastic noise $\mathcal{E}_\tau$ via the backward process:

$$\mathcal{E}_{\tau+1} = \sqrt{\alpha_{\tau+1}} f_\theta(\mathcal{E}_\tau, c, \tau) + \sqrt{1 - \alpha_{\tau+1}} \epsilon_\theta(\mathcal{E}_\tau, c, \tau), \quad (10)$$

where f_θ is the denoising function: $f_\theta(\mathcal{E}_\tau, c, \tau) = 1/\sqrt{\alpha_\tau} (\mathcal{E}_\tau - \sqrt{1 - \alpha_\tau} \epsilon_\theta(\mathcal{E}_\tau, c, \tau))$. Based on this process, we can generate a sample-specific random noise and then incorporate with the condition c to decode new event data with Eq. 7, as shown in Fig. 2 (b). For event data generation, *i.e.*, event data augmentation, we can slightly perturb the stochastic noise or modify the decoder's parameters to diversify the outputs ($\hat{\mathcal{E}}$) as:

$$\hat{\mathcal{E}} = \begin{cases} \text{Generate}(\mathcal{E}_T + \epsilon_s, c, T \rightarrow 0), & \# \text{ stochastic noise} \\ \text{Generate}(\mathcal{E}_{\hat{T}}, c, \hat{T} \rightarrow 0), & \# \text{ diffusion step} \\ \text{Generate}(\mathcal{E}_T, c + \epsilon_c, T \rightarrow 0), & \# \text{ condition noise} \end{cases} \quad (11)$$

where $\text{Generate}(\cdot)$ denotes the generative process of our event decoder, $\hat{T}$ indicates the modified number of diffusion steps, ϵ_s and ϵ_c represent the random noise corresponding to the sampled noise $\mathcal{E}_T$ and our condition c , respectively.

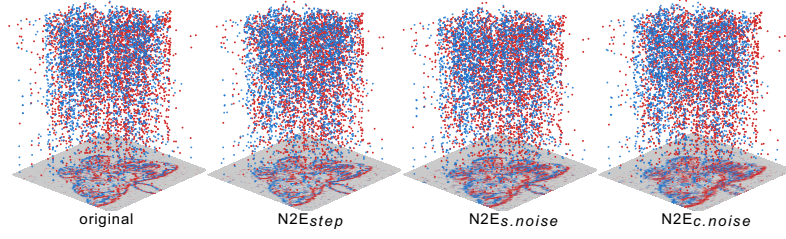


Fig. 3: Original and N2E-augmented event data. N2E generates new event data from noise through three operations: adjusting denoising steps ($N2E_{step}$), perturbing the sampled Gaussian noise ($N2E_{s.noise}$), add adding condition noise ($N2E_{c.noise}$). Blue and red points denote the positive and negative events, respectively.

Based on Eq. 11, we can minimize the diffusion steps to generate event samples with more background noise; perturb the condition c to populate event details; or add noise to the $\mathcal{E}_T$ to blur the object edges, *etc.*

Visualization illustration of Eq. 11 is shown in Fig. 3. Compared to the original event data, the augmented samples demonstrate substantial diversity while preserving the integrity of the main targets. $N2E_{step}$ renders the new event data by varying the number of denoising steps in the event decoder. $N2E_{s.noise}$ produces samples by perturbing the sampled stochastic noise through random erasure or addition of additional noise components. $N2E_{c.noise}$ creates the augmented data by polluting the extracted condition messages. These three operations enable N2E to synthesize diverse event data from noise, thereby effectively alleviating the overfitting issue.

3.3 Implementation details

For N2E, we set the number of diffusion iteration (T) as 1000 and train the diffusion model with 1000 epochs. Adam optimizer is employed with the learning rate of $1e^{-4}$. Δ_n is set to 4096 for event slicing. The batch size is set to 64. For event data augmentation, we set the $T \in [50, 200]$ since the larger T , the less noise. The scaling factors ϵ_s and ϵ_c are 0.1. 30% of the training data are randomly sampled to generate the augmented version. For downstream event vision models, we train each model with the same learning rate and optimizer employed in N2E. The batch size and epoch number of these models are set to 16 and 30, respectively. Note that the parameters of each event representation are directly copied from their official implementations. All models are trained and evaluated on a single NVIDIA A100 GPU.

4 Experiment

4.1 Setup

Dataset. We evaluate our method on two widely used event-based datasets across different tasks. For object classification, we employ N-Caltech101 [34],

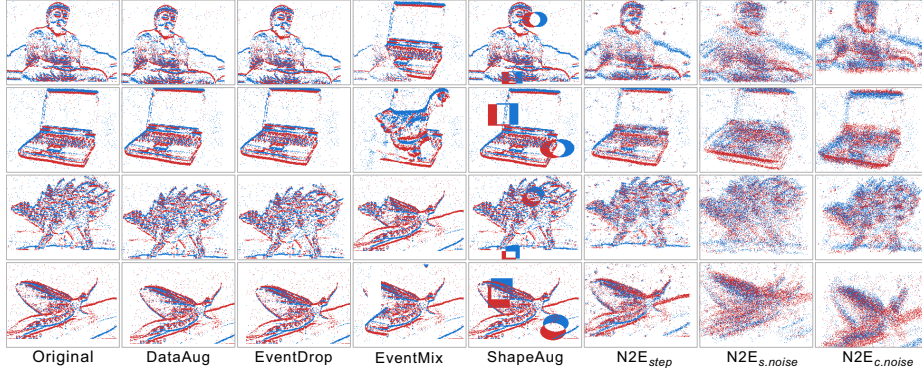


Fig. 4: Qualitative results on N-Caltech101 dataset [34]. $N2E_{step}$, $N2E_{s.noise}$, and $N2E_{c.noise}$ denote the augmented samples by modifying diffusion steps, perturbing stochastic noise, and adding condition noise, respectively.

which contains 8,246 samples across 101 object categories. For action recognition, we use DVS-Gesture [1], such a dataset comprising 1,342 instances across 11 hand-gesture classes performed by 29 subjects under various lighting conditions.

Table 1: Classification accuracy on N-Caltech101 [34] dataset. Best results on each representation are **bolded**. All results are reported in percentage (%).

	<i>Event Frame [31]</i>					<i>Time Surface [25]</i>					<i>Voxel Grid [56]</i>					<i>Event Spike Tensor [9]</i>				
	RN18	RN50	VGG	ViT	Swin	RN18	RN50	VGG	ViT	Swin	RN18	RN50	VGG	ViT	Swin	RN18	RN50	VGG	ViT	Swin
Original	73.7	79.1	64.4	46.2	88.1	75.4	81.8	66.0	38.8	88.7	79.7	85.8	81.2	42.0	91.3	81.5	85.4	72.7	46.7	87.5
DataAug	82.4	83.6	71.1	55.3	88.3	84.3	84.1	74.4	47.4	89.5	85.6	88.4	81.6	61.7	91.4	86.8	86.4	81.4	59.4	88.9
EventDrop	79.3	81.6	62.9	48.3	86.4	82.6	83.9	74.1	42.9	89.7	86.6	87.2	81.1	46.5	91.9	85.2	86.8	78.8	54.9	90.5
EventMix	53.0	76.7	72.8	37.3	86.5	80.2	78.9	74.6	43.9	91.3	83.2	85.5	81.2	46.4	92.0	82.8	83.8	79.8	54.8	74.0
ShapeAug	79.1	80.6	64.8	44.6	86.1	80.5	81.2	36.3	38.1	85.3	84.0	87.0	76.3	36.9	88.7	81.5	85.6	69.9	48.3	89.2
N2E (Ours)	83.1	90.6	74.9	56.1	90.1	88.0	87.7	78.5	50.2	94.3	87.8	90.5	85.2	50.0	94.9	86.5	87.8	82.2	63.0	91.2

Baseline. We compare against four representative augmentation methods: 1) DataAug [9], which applies conventional data augmentation techniques (*e.g.*, rotation, flipping) to event data; 2) EventDrop [12], which stochastically removes events in spatial or temporal dimensions to improve robustness; 3) EventMix [38], which mixes different event streams via a spatiotemporal mask to generate new samples; and 4) ShapeAug [4], which augments event data by translating geometric shapes across the spatial domain.

Metrics. We evaluate all methods using top-1 accuracy on both datasets. To ensure comprehensive evaluation, we conduct experiments with five popular deep learning architectures (ResNet18 (RN18) [14], ResNet50 (RN50) [14], VGG16 (VGG) [41], ViT [7], Swin [32]) combined with four widely-adopted event representations (Event Frame [31], Time Surface [25], Voxel Grid [56], and Event Spike Tensor [9]). The mean accuracy across all network-representation configurations is reported for each augmentation method.

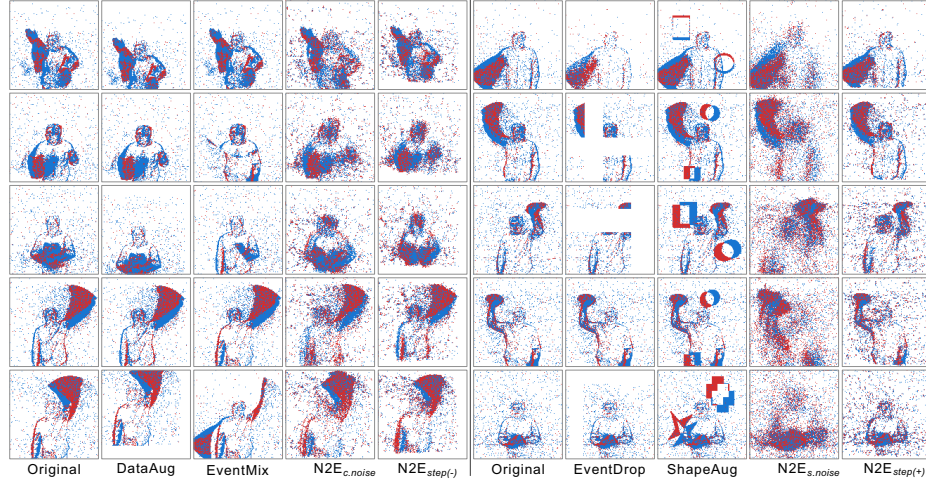


Fig. 5: Qualitative comparisons of different methods on DVS-Gesture dataset [1]. $N2E_{step}$, $N2E_{s.noise}$, and $N2E_{c.noise}$ denote the augmented samples by modifying diffusion steps, adding stochastic noise, and adding condition noise, respectively. “-” and “+” denote the smaller and larger diffusion step numbers, respectively.

4.2 Object classification

Qualitative comparison. Fig. 4 presents a qualitative comparison among representative event data augmentation methods, highlighting the significant differences in diversity of generated event streams. DataAug [9] mainly employ the traditional data translation (*e.g.*, translations, rotations) to diversify the spatial dimensions. As shown in the 2nd column of Fig. 4, the generated samples of DataAug [9] remain nearly identical to the original ones, leading to limited diversity. EventDrop [12] randomly removes a subset of events, partially simulates missing-event scenarios. However, this operation destroys temporal continuity, resulting in corrupted representations that can impair the convergence of downstream models. EventMix [38] combines two event streams by a 3D mask, thereby increasing variation in spatiotemporal patterns. While this strategy enhances diversity, the visualizations reveal excessive overlap, yielding unrealistic and ambiguous patterns that deviate from physical capture, as shown in the 4th column. ShapeAug [4] introduces simulated event data at the object level to augment the event data, showing limited diversity.

By modeling the distribution of event data, N2E generates coherent event streams with diverse background noise and blurry objects. The examples shown in the last three columns of Fig. 4 are generated with different diffusion configurations, such as various diffusion steps, perturbed stochastic noise and condition noise. These examples indicate that diffusion-based augmentation offers a more principled mechanism for generating diverse yet physically valid event data. Unlike transformation- or rule-based methods that perturb existing samples, N2E

Table 2: Action recognition performance on DVS-Gesture [1] dataset. Best results on each representation are **bolded**. All results are reported in percentage (%).

	<i>Event Frame [31]</i>					<i>Time Surface [25]</i>					<i>Voxel Grid [56]</i>					<i>Event Spike Tensor [9]</i>				
	RN18	RN50	VGG	ViT	Swin	RN18	RN50	VGG	ViT	Swin	RN18	RN50	VGG	ViT	Swin	RN18	RN50	VGG	ViT	Swin
Original	77.8	84.0	77.4	62.5	58.3	74.3	69.8	69.8	47.6	64.6	87.5	89.9	81.6	63.9	74.3	86.1	88.5	82.6	77.4	78.5
DataAug	85.4	85.8	86.8	73.3	62.9	80.6	83.3	79.9	53.1	69.8	93.8	94.4	84.4	82.3	87.5	89.6	90.3	81.6	84.0	78.8
EventDrop	83.7	84.0	81.9	70.1	72.9	—	—	—	—	—	92.0	92.7	89.2	80.2	16.7	87.2	69.1	83.0	80.9	82.3
EventMix	84.0	81.6	85.1	65.3	50.7	71.2	67.4	77.4	37.5	61.1	87.9	90.3	87.9	81.6	79.2	86.5	87.5	85.1	77.1	83.3
ShapeAug	77.8	83.3	78.5	63.5	61.8	72.2	73.3	75.7	44.4	66.7	89.9	92.0	85.4	59.4	71.9	87.5	88.9	77.1	69.4	78.1
N2E (Ours)	85.1	88.5	88.2	77.8	73.3	82.6	86.1	80.6	55.9	70.1	97.6	96.2	91.7	83.7	93.4	89.2	91.0	85.8	89.6	84.7

“—” indicates training failed to converge with NaN loss.

learns to reconstruct plausible event distributions directly from data. This enables it to improve both the fidelity and variability of augmented events.

Quantitative comparison. Table 1 presents the classification accuracy of five deep models among four event representations on the N-Caltech101 [34] dataset. Four event data augmentation baselines are employed. Our N2E achieves a competitive performance across different network architectures. Compared to original results (without augmentation), N2E brings substantial improvements, with gains ranging from 2% to over 11%. DataAug [9] raises the accuracy by about 5–7%. EventDrop [12] offers moderate gains on ResNets [14] but degrades ViT [7] and VGG [41] results. EventMix [38] shows inconsistent improvements due to the ambiguous label assignment, such as improving the accuracy of VGG [41] network from 64.4% to 72.8% while dropping the accuracy of the ResNet18 [14] from 73.7% to 53.0% among the Event Frame [31]. ShapeAug [4] performs comparably to the baseline, delivering minimal gains except on select models. Notably, the Swin Transformer [32] with N2E reaches 94.9% using the Voxel Grid [56] and 94.3% with the Time Surface [25], both the highest accuracies among all combinations. Overall, N2E demonstrates the high effectiveness across diverse architectures and event representations on the classification task.

4.3 Action recognition

Qualitative comparison. The visualization results in Fig. 5 provide a comprehensive qualitative comparison of various event data augmentation methods on the DVS-Gesture [1] dataset. The DataAug [9] and EventDrop [12] methods exhibit moderate visual quality but also expose key limitations inherent to traditional augmentation schemes. EventDrop [12] suffers from excessive sparsity due to random event removal, leading to incomplete gesture representations. These issues suggest that such methods can introduce unrealistic variations that may harm downstream network generalization when overapplied. EventMix [38] and ShapeAug [4] attempt to synthesize new samples through introducing new event streams. EventMix [38] yields plausible structures but sometimes generates unnatural overlaps. By contrast, the proposed N2E generates new event data with noticeable blurry and background noise. The generated samples mainly reproduce the spatial distribution and temporal dynamics of real events, maintaining

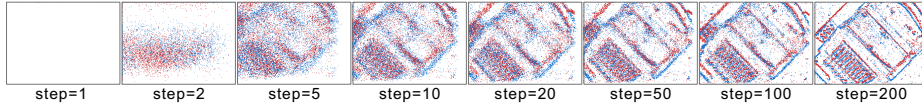


Fig. 6: Denoising ability of N2E with different diffusion steps.

coherent gesture outlines and realistic event densities. By doing so, users can generate various gesture event data with blurry and background noise.

Quantitative comparison. In Table 2, we evaluate action recognition accuracy of five deep models on the DVS-Gesture [1] dataset with and without event augmentation baselines. Among four comparison methods, our N2E achieves the highest top-1 accuracy on almost all cases. DataAug [9] shows relatively stable improvement for different downstream models among four event representations. For example, the accuracies of ResNet18 under the event frame and event spike tensor are improved from 77.8% to 85.4%, and 86.1% to 89.6%, respectively. EventDrop [12] randomly drops events to augment new data but shows failures on Time Surface [25], resulting in poor model convergence (NaN loss). EventMix [38], mixing event streams with interpolated labels, offers modest improvements for voxel grid and spike tensor representations. ShapeAug [4], introducing synthetic occlusions, achieves minor gains than original training. Only a 0.4% improvement is achieved for ResNet50 when employing ShapeAug under the event spike tensor representation. Totally, N2E outperforms four comparison baselines on almost all cases, delivering the largest gains. The accuracies of ViT and Swin under the voxel grid are improved from 63.9% and 74.3% to 83.7% and 93.4%, respectively. Our N2E demonstrates superior adaptability to different downstream models and various event representations.

4.4 Discussion

Denoising ability of N2E. Fig. 6 illustrates the progressive denoising performance of N2E as the number of diffusion steps increases from 1 to 200. At $step = 1$, the generated frame is nearly blank, indicating that the model starts from a noise-dominant distribution without meaningful structure. At $step = 2 \sim 5$, random sparse activations begin to appear, but the event pattern remains highly stochastic and lacks coherent object contours. By $step = 10$, the underlying geometric structure of the target object begins to emerge, suggesting that the diffusion model has begun to recover salient spatial correlations from the noise. At $step = 20$, the object edges become markedly clearer, and the event polarity (indicated by red and blue points) aligns with the object’s actual spatiotemporal gradient, reflecting effective denoising. With further increments to $step = 50, 100, 200$, fine-grained structural details continue to refine, while background noise is substantially reduced. The event patterns become dense and consistent with a plausible event frame, implying convergence of the generative

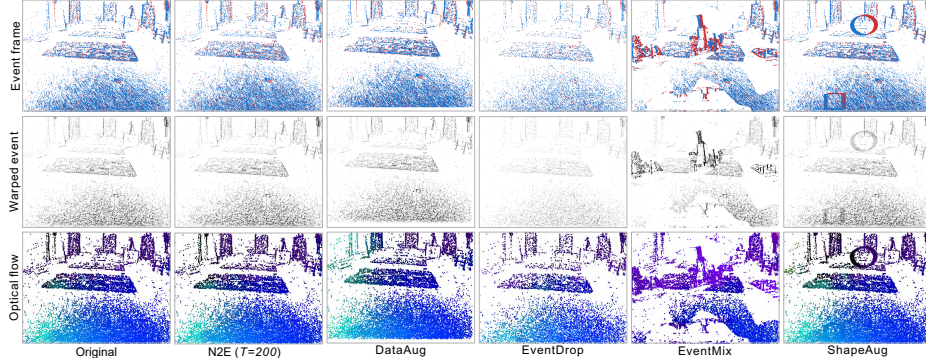


Fig. 7: Visual comparison of different event data augmentation methods on MVSEC dataset [55]. For event frame, Blue and red points denote the positive and negative event data, respectively. Notably, our N2E shows high flow consistency with the original data that effectively demonstrates reconstruction ability, leading a solid foundation for following in-distribution data generation.

process. From this observation, we can draw the conclusion that N2E gradually reconstructs meaningful event distributions as the diffusion process proceeds. The network learns how to reverse the noise corruption and recover the event data. Consequently, tuning the number of diffusion steps provides a controllable trade-off: fewer steps generate more stochastic and noisy data for augmentation, while more steps yield high-fidelity reconstructions.

Fidelity of N2E. To generate in-distribution event data, we must ensure that N2E accurately captures spatial, temporal, and polarity characteristics of our target data. Evaluating the optical flow amps of generated data provides an effective validation approach, which is because the flow information indicates the specific motion process between two consecutive events. Thus, we illustrate event frames, warped events, and optical flow maps in Fig. 7.

The event frames are shown in the 1st row of Fig. 7, which visualize spatial-temporal event distributions with polarity information (blue and red points). N2E generates event frames nearly indistinguishable from ground truth, maintaining natural event density and polarity distribution. In contrast, EventDrop exhibits visible sparsity due to intentionally dropping events by random time stamps. EventMix randomly mix two events with a mask that displays problematic artifacts. ShapeAug introduces simulated events to diversify the original event data, which may not reflect natural scene dynamics.

Optical Flow maps are depicted in the last row of Fig. 7, where color indicates direction and intensity represents magnitude. N2E generates flow fields that closely match ground truth in terms of color distribution and magnitude patterns. DataAug [9] shows different flow patterns while adopting the simple coordinate translation. EventDrop [12] maintains flow structure despite reduced density, EventMix [38] produces severe flow artifacts with color inconsistencies,

Table 3: Quantitative evaluation of different event augmentation methods on MVSEC dataset [55]. N2E with $T = 200$ achieves superior data quality compared to existing baselines, demonstrating its strong capability in modeling the original data distribution.

	DataAug	EventDrop	EventMix	ShapeAug	N2E
ePSNR $\uparrow$	2.1627	2.1587	1.2190	2.1356	11.9747
RMSE $\downarrow$	0.3694	0.3700	0.5704	0.3740	0.0040
LPIPS $\downarrow$	0.6342	0.6156	0.6839	0.6355	0.1910

spatial discontinuities that break motion coherence. Although ShapeAug 4 injects simulated event data that doesn’t affect the original optical flows, the simulated object shows false flows due to the confused time stamps.

Overall, N2E demonstrates clear superiority through its physical consistency across spatial, temporal, and polarity dimensions that can generate in-distribution data indistinguishable from real events. The warped events (2nd row in Fig. 7) also show that N2E achieves excellent motion compensation with clean warped events comparable to ground truth. This validates the physical plausibility of the generated data. And the quantitative results in Table 3 demonstrate the accurate generation ability of the N2E. To evaluate the difference between the original and augmented event data, we test the PSNR and root mean square error at the event-level [51], denoted as ePSNR and RMSE, respectively. The LPIPS is calculated on the event frame-level [31]. In Table 3, N2E achieves the best ePSNR (11.9747), RMSE (0.004), and LPIPS (0.191), which effectively demonstrates its ability in in-distribution data generation. Please refer to the Supplementary Material for more details and ablation studies.

5 Conclusion

In this work, we present N2E, a diffusion-based event data augmentation framework that generates diverse event data from the sampled noise. N2E integrates an event decoder and a condition extractor to render the input event data with various blurry and background noise. A dimension-decoupled event representation method is specifically designed for N2E to model the event data distribution. Extensive experiments on two downstream tasks among five event vision models demonstrate that N2E mitigates the overfitting issue more effectively than existing baselines. These results highlight the potential of generation-based strategies for event data augmentation from the noise.

Limitation. N2E utilizes a diffusion-based generative process to synthesize event data via progressive denoising. As a result, its efficiency is lower than that of mask- or mixing- based baselines because of the iterative denoising procedure. Re-running the reverse process of the event decoder to sample Gaussian noise is another factor that degrades the efficiency. Therefore, we plan to improve the efficiency through presenting the accelerated denoising procedure and a simplified noise sampling strategy in the future, thereby enhancing the applicability of N2E across various event-based vision tasks.

Acknowledgment. This work was carried out at the Renjie Group, Hong Kong Baptist University. Renjie Group was supported by the National Natural Science Foundation of China under Grant No. 62302415, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2024A1515012822, and the Research Grant Council (RGC) of Hong Kong SAR, China, under a GRF Grant 12203124 and an ECS Grant 22201125. Ziyuan Luo was supported by a fellowship award from the RGC of Hong Kong SAR, China (Project No. HKBU JRF52627-2S03). Peiqi Duan and Boxin Shi were supported by Beijing Natural Science Foundation (Grant No. L233024), Beijing Municipal Science & Technology Commission, Administrative Commission of Zhongguancun Science Park (Grant No. Z241100003524012), and Beijing High Innovation Plan (Contract No. 202604841443). Peiqi Duan was also supported by National Natural Science Foundation of China (Grant No. 62402014), China National Postdoctoral Program for Innovative Talents (Grant No. BX20230010), and China Postdoctoral Science Foundation (Grant No. 2023M740076).

References

1. Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al.: A low power, fully event-based gesture recognition system. In: Proc. CVPR (2017)
2. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R.: Structured denoising diffusion models in discrete state-spaces. In: Proc. NeurIPS (2021)
3. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proc. CVPR (2022)
4. Bendig, K., Schuster, R., Stricker, D.: ShapeAug: Occlusion augmentation for event camera data. arXiv preprint arXiv:2401.02274 (2024)
5. Chakravarthi, B., Verma, A.A., Daniilidis, K., Fermuller, C., Yang, Y.: Recent event camera innovations: A survey. In: Proc. ECCVW (2024)
6. Dong, Y., He, X., Shen, G., Zhao, D., Li, Y., Zeng, Y.: EventZoom: A progressive approach to event-based data augmentation for enhanced neuromorphic vision. In: Proc. AAAI (2025)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. ICLR (2020)
8. Duan, P., Wang, Z.W., Zhou, X., Ma, Y., Shi, B.: Eventzoom: Learning to denoise and super resolve neuromorphic events. In: Proc. CVPR (2021)
9. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proc. ICCV (2019)
10. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: DSEC: A stereo event camera dataset for driving scenarios. RA-L **6**(3), 4947–4954 (2021)
11. Gu, F., Dou, J., Li, M., Long, X., Guo, S., Chen, C., Liu, K., Jiao, X., Li, R.: EventAugment: learning augmentation policies from asynchronous event-based data. TCDS **16**(4), 1521–1532 (2024)
12. Gu, F., Sng, W., Hu, X., Yu, F.: EventDrop: Data augmentation for event-based learning. In: Proc. IJCAI (2021)
13. Han, J., Zhou, C., Duan, P., Tang, Y., Xu, C., Xu, C., Huang, T., Shi, B.: Neuro-morphic camera guided high dynamic range imaging. In: Proc. CVPR (2020)

14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR (2016)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Proc. NeurIPS (2020)
16. Huang, Z., Sun, L., Zhao, C., Li, S., Su, S.: EventPoint: Self-supervised interest point detection and description for event-based camera. In: Proc. WACV (2023)
17. Jain, S., Watson, D., Tabellion, E., Poole, B., Kontkanen, J., et al.: Video interpolation with diffusion models. In: Proc. CVPR (2024)
18. Jiang, J., Li, J., Zhang, B., Deng, X., Shi, B.: EvHandPose: Event-based 3d hand pose estimation with sparse supervision. TPAMI **46**(9), 6416–6430 (2024)
19. Jiang, Z., Zhang, Y., Zou, D., Ren, J., Lv, J., Liu, Y.: Learning event-based motion deblurring. In: Proc. CVPR (2020)
20. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Proc. CVPR (2023)
21. Kim, J., Bae, J., Park, G., Zhang, D., Kim, Y.M.: N-imagenet: Towards robust, fine-grained object recognition with event cameras. In: Proc. ICCV (2021)
22. Kim, J., Hwang, I., Kim, Y.M.: Ev-TTA: Test-time adaptation for event-based object recognition. In: Proc. CVPR (2022)
23. Kim, T., Cho, H., Yoon, K.J.: Frequency-aware event-based video deblurring for real-world motion blur. In: Proc. CVPR (2024)
24. Klenk, S., Chui, J., Demmel, N., Cremers, D.: TUM-VIE: The tum stereo visual-inertial event dataset. In: Proc. IROS (2021)
25. Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B.: HOTS: a hierarchy of event-based time-surfaces for pattern recognition. TPAMI **39**(7), 1346–1359 (2016)
26. Li, H., Liu, H., Ji, X., Li, G., Shi, L.: CIFAR10-DVS: an event-stream dataset for object classification. Front. Neurosci. **11** (2017)
27. Li, X., Ren, Y., Jin, X., Lan, C., Wang, X., Zeng, W., Wang, X., Chen, Z.: Diffusion models for image restoration and enhancement: A comprehensive survey. IJCV **133**(11), 8078–8108 (2025)
28. Li, Y., Kim, Y., Park, H., Geller, T., Panda, P.: Neuromorphic data augmentation for training spiking neural networks. In: Proc. ECCV (2022)
29. Liang, J., Yang, Y., Li, B., Duan, P., Xu, Y., Shi, B.: Coherent event guided low-light video enhancement. In: Proc. ICCV (2023)
30. Liu, H., Chen, G., Qu, S., Zhang, Y., Li, Z., Knoll, A., Jiang, C.: TMA: Temporal motion aggregation for event-based optical flow. In: Proc. ICCV (2023)
31. Liu, M., Delbruck, T.: Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In: Proc. BMVC (2018)
32. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. ICCV (2021)
33. Millerdurai, C., Akada, H., Wang, J., Luvizon, D., Theobalt, C., Golyanik, V.: EventEgo3D: 3d human motion capture from egocentric event streams. In: Proc. CVPR (2024)
34. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Front. Neurosci. **9**, 437 (2015)
35. Preechakul, K., Chatthee, N., Wizadwongsa, S., Suwajanakorn, S.: Diffusion autoencoders: Toward a meaningful and decodable representation. In: Proc. CVPR (2022)

36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proc. CVPR (2022)
37. Schaefer, S., Gehrig, D., Scaramuzza, D.: AEGNN: Asynchronous event-based graph neural networks. In: Proc. CVPR (2022)
38. Shen, G., Zhao, D., Zeng, Y.: EventMix: An efficient data augmentation strategy for event-based learning. *Information Sciences* **644**, 119170 (2023)
39. Shiba, S., Aoki, Y., Gallego, G.: Secrets of event-based optical flow. In: Proc. ECCV (2022)
40. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *Journal of Big Data* **6**(1), 1–48 (2019)
41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc. ICLR (2015)
42. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: Proc. ICLR (2021)
43. Sun, M., Zhang, D., Ge, Z., Jiaxu, W., Li, J., Fang, Z., Xu, R.: EventRPG: Event data augmentation with relevance propagation guidance. In: Proc. ICLR (2024)
44. Sun, Z., Messikommer, N., Gehrig, D., Scaramuzza, D.: ESS: Learning event-based semantic segmentation from still images. In: Proc. ECCV (2022)
45. Wang, R., Duan, P., Cheung, K.C., See, S., Shi, B., Wan, R.: Texvent: Asynchronous event data simulation via text prompt. In: Proc. CVPR (2026)
46. Wang, R., Duan, P., Shi, B., Wan, R.: Asynchronous event error-minimizing noise for safeguarding event dataset. In: Proc. ICCV (2025)
47. Wang, R., Guo, Q., Li, H., Wan, R.: Event Trojan: Asynchronous event-based backdoor attacks. In: Proc. ECCV (2024)
48. Xia, B., Zhang, Y., Wang, S., Wang, Y., Wu, X., Tian, Y., Yang, W., Van Gool, L.: DiffIR: Efficient diffusion model for image restoration. In: Proc. ICCV (2023)
49. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: Regularization strategy to train strong classifiers with localizable features. In: Proc. ICCV (2019)
50. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: MixUp: Beyond empirical risk minimization. In: Proc. ICLR (2018)
51. Zhang, Z., Cui, S., Chai, K., Yu, H., Dasgupta, S., Mahbub, U., Rahman, T.: V2CE: Video to continuous events simulator. In: Proc. ICRA (2024)
52. Zhou, C., Teng, M., Han, J., Xu, C., Shi, B.: Delieve-net: Deblurring low-light images with light streaks and local events. In: Proc. ICCV (2021)
53. Zhou, X., Duan, P., Xiaokaiti, Y., Xu, C., Shi, B.: Event-based visual vibrometry. In: Proc. ICCV (2025)
54. Zhou, Z., Shao, S., Bai, L., Zhang, S., Xu, Z., Han, B., Xie, Z.: Golden noise for diffusion models: A learning framework. In: Proc. ICCV (2025)
55. Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *RA-L* **3**(3), 2032–2039 (2018)
56. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proc. CVPR (2019)
57. Zubić, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: From chaos comes order: Ordering event representations for object recognition and detection. In: Proc. ICCV (2023)

From Noise to Events: Conditional Diffusion for Event Data Augmentation

Supplementary Material

Ruofei Wang¹, Ziyuan Luo¹, Peiqi Duan², Xiufeng Huang¹,
Boxin Shi², and Renjie Wan^{1*}

¹ Department of Computer Science, Hong Kong Baptist University

² School of Computer Science, Peking University

{ruofei, ziyuanluo, xiufenghuang}@life.hkbu.edu.hk, {duanqi0001, shiboxin}@pku.edu.cn, renjiewan@hkbu.edu.hk

In this supplementary material, we present a detailed discussion of N2E and additional experiments. These experiments include qualitative evaluations of three augmentation operations, analysis of cross-sample generation capabilities, generalization performance on event object detection and large-scale datasets, computational efficiency benchmarks, and comparisons with additional baselines.

1 About N2E

Currently, N2E synthesizes new event data by perturbing the sampled noise and condition vectors, or modifying denoising steps. N2E fails to directly generate event data from the random Gaussian noise due to two factors. First, the inherent sparsity and asynchronous nature of event data fundamentally differ from dense pixel-based RGB frames (see Fig. 1), making it significantly harder for diffusion models to learn a robust data distribution from random noise. Second, the scalability of the event datasets remains limited. The datasets employed in the current event dataset contain only a few thousand samples, which is much smaller than the datasets used to train state-of-the-art image diffusion models. This scarcity restricts the model’s ability to fully capture the rich, high-dimensional distribution of event data, leading to poor generalization performance when generating events from pure noise. Thus, the hybrid dataset training (*i.e.*, enlarging the scalability of the event dataset) and novel event representation strategies (*i.e.*, minimizing the difference between the event data and RGB images) require deeper exploration in the future.

2 Additional experiments

2.1 Importance of augmentation operations

To generate new event data, we present employing three operations during the inference stage, as described in Eq. 11. We evaluate the impact of each oper-

² Peiqi Duan and Boxin Shi are with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University. *Corresponding author.

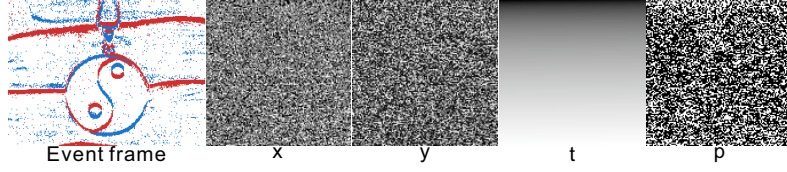


Fig. 1: Event frame and 4-channel representation of the event data. x , y , t , and p are normalized to $[0, 1]$, respectively.

ation on both event quality and downstream performance Table 1. ePSNR denotes the event-level peak signal-to-noise ratio, which is calculated as: $\text{ePSNR} = 10 \times \log_{10}(\max(\mathcal{E}_1)) / \sqrt{(\mathcal{E}_1 - \mathcal{E}_2)^2}$. $\mathcal{E}$ indicates the event data. RMSE is the root mean square error between the original and generated event data. LPIPS and ACC denote the learned perceptual image patch similarity and accuracy, respectively. In Table 1, varying the number of diffusion steps (E1) introduces less noise compared to the other two operations. Perturbing the sampled noise $\mathcal{E}_T$ (E2) results in lower event quality than adding noise to the condition c (E3). Regarding downstream performance, modifying the number of diffusion steps and injecting condition noise yield better improvements than perturbing $\mathcal{E}_T$. This is because $\mathcal{E}_T$ releases more influence for the content of the final diffused event data. Randomly applying all three operations (E4) achieves the highest accuracy of 0.9058, representing an improvement of 0.1146 over the baseline (E0).

Table 1: Ablation study about each operation on the downstream task: object classification. ResNet50 [6] is evaluated on the N-Caltech101 dataset [10] with the representation approach: Event Frame [9].

T	$\mathcal{E}_T$	c	ePSNR	RMSE	LPIPS	ACC
E0			11.9747	0.0040	0.0190	0.7912
E1	✓		9.4352	0.0130	0.2944	0.9031
E2	✓		2.1217	0.3764	0.3717	0.8470
E3		✓	2.2932	0.3478	0.2566	0.8962
E4	✓	✓	5.9346	0.1851	0.2883	0.9058

2.2 Effectiveness on cross-sample generation

To generate new event data, we implement three operations, including adjusting diffusion steps, perturbing the sampled stochastic noise, and polluting the condition vectors. For the latter two operations, we can draw noise from other event sequences to explore cross-sample effectiveness. Alternative solutions for the last two operations are sampling noise from other event data. Fig. 2 illustrates the visualization examples under different pairings of the condition vector

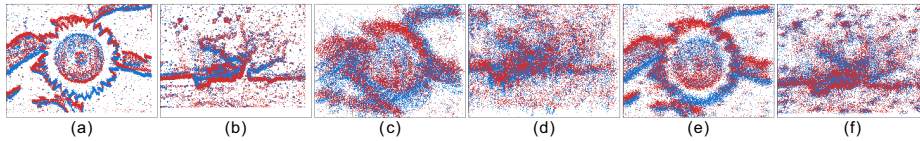


Fig. 2: Visualization of augmented event data. (a) and (b) show cleanly generated event data for Sunflower and Joshua Tree, respectively. (c) is conditioned on Joshua Tree with random noise sampled from Sunflower; (d) shows the converse. (e) is conditioned on Sunflower with random noise generated from both Sunflower and Joshua tree (mixing parameter: 1 and 0.05); (f) represents the converse.

and the sampled noise. Case (a) and (b) depict the event data generated by N2E with condition vectors and sampled noise from Sunflower and Joshua tree itself, respectively. Cross-pairing the Joshua Tree condition with the sampled noise of Sunflower (c) produces an event that resembles the Sunflower pattern but with blurred details; the reverse pairing (d) yields a Joshua Tree-like structure with reduced sharpness. When we blend the noise sampled from both sequences (with mixing parameters) while keeping the original condition vectors (e, f), the generated data retain the expected global pattern while exhibiting richer background events. Consequently, we can augment new event data by sourcing condition vectors and stochastic noise from other samples rather than sampling them randomly. Future work will explore additional augmentation strategies in N2E to further diversify and enrich event datasets.

2.3 Generalizability on event-based object detection

We conduct an experiment on DSEC-Det [3] with Yolov5 [11] to evaluate our effectiveness of our N2E. The popular object detection metrics: Precision (P), Recall (R), and mAP, are employed. As shown in Table 2, N2E effectively improves the P and mAP of the original detector (w/o N2E) from 0.68 and 0.30 to 0.85 and 0.33, respectively. Such a comparison demonstrates that N2E not only benefits the classification and recognition tasks, but also shows high generalizability for other event-based domains.

Table 2: N2E on DSEC-Det.

	P	R	mAP _{.5:.95}
w/o N2E	0.68	0.48	0.30
w/ N2E	0.85	0.43	0.33

Table 3: N2E on N-Imagenet.

	Accuracy
w/o N2E	47.0%
w/ N2E	48.0%

2.4 Influence of dimension-decoupled event representation

Current representation methods mainly compress dense timestamps into fixed intervals. Thus, reconstructing raw data from such representations is challenging.

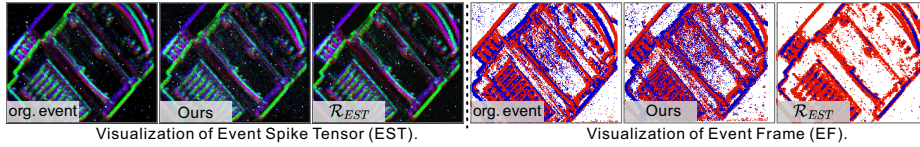


Fig. 3: Visualization comparison of our dimension-decoupled representation and Event Spike Tensor [2]. When our representation method is used during training, the augmented data remains visually consistent with the original data even using other representations (*e.g.*, EST [2], EF [9]). In contrast, when EST [2] is employed during training, the data represented through EF [9] exhibits substantial discrepancies from the original data.

In contrast, our dimension-decoupled approach divides an event stream into multiple segments and then directly reshapes them into a 4-channel map according to coordinate, time stamp and polarity, without information loss. Such a dimension-decoupled representation method is specifically designed for N2E to ensure the temporal fidelity of augmented samples. We position this representation as an enabling component of N2E for augmentation rather than a general event representation for other tasks. Existing representation strategies (*e.g.*, Event Spike Tensor [2], Event Frame [9]) often compress dense timestamps, which hinders the direct augmentation of raw data. To solve this, we propose a new representation that directly introduces blurriness into the data while performing consistently for various representations, as shown in Fig. 3. Training with EST ($\mathcal{R}_{EST}$) produces data that performs well in EST (left) but poorly in EF (right). Our method, however, achieves high quality in both EST and EF representations.

2.5 Efficiency evaluation

The runtime of each augmentor is shown in Table 4. DataAug [2] shows the best efficiency due to its simple operation (*e.g.*, flipping). Among other comparison baselines, EventMix [12] takes much time due to the 3D mask initialization and label allocation. N2E takes almost 7.6s to augment an event data, which is mainly caused by the highly computational step-by-step denoising process.

Table 4: Runtime evaluation.

	DataAug	EventDrop	EventMix	ShapeAug	N2E
Runtime	1.0 ms	3.6 ms	11.6 ms	5.9 ms	7.6 s

2.6 Effectiveness on N-Imagenet

We evaluate our N2E on N-Imagenet and show results in Table 3. We sample ten classes for this evaluation due to the training burden. As shown in Table 3,

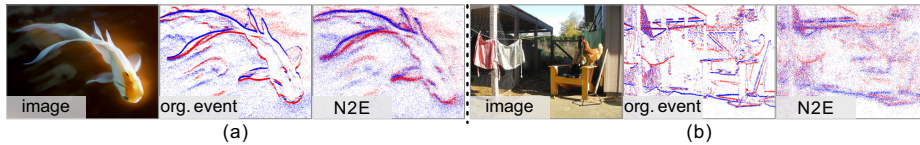


Fig. 4: Visualization results of N2E on N-imagenet dataset [7]. (a) N2E produces high-quality event data that aligns well with the original when the proportion of activated background events is low. (b) Conversely, when background noise events constitute the majority of the data, N2E cannot properly capture the target objects (*i.e.*, hen).

Table 5: Results on N-Caltech101 under representations of Event Spike Tensor (EST) [2] and Voxel Grid (VG) [14].

	EventRPG		NDA		EventAugment		N2E	
	EST	VG	EST	VG	EST	VG	EST	VG
Acc	85.41	85.21	83.56	83.52	87.61	/	87.83	90.52

our N2E can improve the original accuracy (Org.) from 47% to 48% (Aug.). The improvement is limited because the original data in N-Imagenet contains considerable noise, which hampers N2E to generate high-quality target objects, as shown in Fig. 4. When background noise events are minimal, N2E generates high-quality event data that closely matches the original (case (a)). However, in case (b), when background noise dominates the data, N2E fails to accurately model the target objects (*i.e.*, hen).

2.7 Additional comparison baselines

In our main experiments, we employ DataAug [2], EventDrop [5], EventMix [12], and ShapeAug [1] as comparison baselines to demonstrate the effectiveness of our N2E. Apart from these four most popular augmentation methods, EventRPG [13], NDA [8], and EventAugment [4] are tested in this section. Similar to EventMix [12], EventRPG [13] augments event data via the data mixture according to a salience map. NDA [8] employs the geometric augmentations (*i.e.*, fixed data transformation) from image fields to augment event data. EventAugment [4] is an improved version of EventDrop [5], which employs an optimization method to select detailed operations of EventDrop [5] to augment event data. Quantitative results of EventRPG [13], NDA [8], and EventAugment [4] are shown in Table 5. The results for EventAugment [4] are taken directly from their paper, as the code has not been released. Compared to EventRPG [13], N2E achieves improvements by 2.42% and 5.31% under representations of EST [2] and VG [14], respectively. Overall, our N2E achieves the best accuracy across both EST [2] and Voxel Grid (VG) [14], demonstrating its effectiveness.

References

1. Bendig, K., Schuster, R., Stricker, D.: ShapeAug: Occlusion augmentation for event camera data. arXiv preprint arXiv:2401.02274 (2024)
2. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
3. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: Dsec: A stereo event camera dataset for driving scenarios. IEEE Robotics and Automation Letters **6**(3), 4947–4954 (2021)
4. Gu, F., Dou, J., Li, M., Long, X., Guo, S., Chen, C., Liu, K., Jiao, X., Li, R.: Eventtaugment: learning augmentation policies from asynchronous event-based data. IEEE Transactions on Cognitive and Developmental Systems **16**(4), 1521–1532 (2024)
5. Gu, F., Sng, W., Hu, X., Yu, F.: Eventdrop: Data augmentation for event-based learning. In: International Joint Conference on Artificial Intelligence (2021)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2016)
7. Kim, J., Bae, J., Park, G., Zhang, D., Kim, Y.M.: N-imagenet: Towards robust, fine-grained object recognition with event cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
8. Li, Y., Kim, Y., Park, H., Geller, T., Panda, P.: Neuromorphic data augmentation for training spiking neural networks. In: European Conference on Computer Vision (2022)
9. Liu, M., Delbruck, T.: Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In: The British Machine Vision Conference (2018)
10. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience **9**, 437 (2015)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
12. Shen, G., Zhao, D., Zeng, Y.: Eventmix: An efficient data augmentation strategy for event-based learning. Information Sciences **644**, 119170 (2023)
13. Sun, M., Zhang, D., Ge, Z., Jiaxu, W., Li, J., Fang, Z., Xu, R.: Eventrpg: Event data augmentation with relevance propagation guidance. In: International Conference on Learning Representations (2024)
14. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2019)