

I2E: Real-Time Image-to-Event Conversion for High-Performance Spiking Neural Networks

Ruichen Ma¹, Liwei Meng¹, Guanchao Qiao¹, Ning Ning¹, Yang Liu¹, Shaogang Hu^{1,2,*}

¹School of Integrated Circuit Science and Engineering, University of Electronic Science and Technology of China

²Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China
{ruichen.ma, lw_meng}@std.uestc.edu.cn, {gcqiao, ning_ning, yliu1975, sghu}@uestc.edu.cn

Abstract

Spiking neural networks (SNNs) promise highly energy-efficient computing, but their adoption is hindered by a critical scarcity of event-stream data. This work introduces I2E, an algorithmic framework that resolves this bottleneck by converting static images into high-fidelity event streams. By simulating microsaccadic eye movements with a highly parallelized convolution, I2E achieves a conversion speed over 300x faster than prior methods, uniquely enabling on-the-fly data augmentation for SNN training. The framework’s effectiveness is demonstrated on large-scale benchmarks. An SNN trained on the generated I2E-ImageNet dataset achieves a state-of-the-art accuracy of 60.50%. Critically, this work establishes a powerful sim-to-real paradigm where pre-training on synthetic I2E data and fine-tuning on the real-world CIFAR10-DVS dataset yields an unprecedented accuracy of 92.5%. This result validates that synthetic event data can serve as a high-fidelity proxy for real sensor data, bridging a long-standing gap in neuromorphic engineering. By providing a scalable solution to the data problem, I2E offers a foundational toolkit for developing high-performance neuromorphic systems. The open-source algorithm and all generated datasets are provided to accelerate research in the field.

Code & Datasets — <https://github.com/Ruichen0424/I2E>

Introduction

Spiking neural networks (SNNs) represent a promising computational paradigm inspired by the brain’s sparse, event-driven processing principles (Xu et al. 2018; Zenke et al. 2021). This bio-inspired design offers a path toward exceptional energy efficiency (Roy, Jaiswal, and Panda 2019; Pei et al. 2019; Zhang et al. 2020; Subbulakshmi Radhakrishnan et al. 2021). When deployed on specialized neuromorphic hardware like Loihi (Davies et al. 2018) or TrueNorth (Merolla et al. 2014), SNNs can achieve orders-of-magnitude gains in power efficiency over conventional artificial neural networks (ANNs), making them ideal candidates for deployment on power-constrained edge devices.

The natural input for an SNN is a stream of asynchronous events, data typically captured by specialized hardware such

Dataset	Architecture	Method	Acc.%
ES-ImageNet (Lin et al. 2021)	ResNet18+LIF	baseline	39.89
	ResNet18+LIF	pre-train	43.74
	ResNet18+LIAF	pre-train	52.25
	ResNet34+LIF	baseline	43.42
	ResNet34+LIAF	pre-train	51.83
N-ImageNet (Kim et al. 2021)	ResNet34	EH	47.73
	ResNet34	STS	47.90
	ResNet34	DiST	48.43
	ResNet34	EST	48.93
I2E-ImageNet (This work)	ResNet18+LIF	baseline-I	48.30
	ResNet18+LIF	baseline-II	57.97
	ResNet18+LIF	pre-train	59.28
	ResNet34+LIF	baseline-II	60.50

Table 1: State-of-the-art comparison on event-based ImageNet classification. The proposed I2E-ImageNet enables an MS-ResNet34 architecture to achieve a new state-of-the-art accuracy of 60.50%, substantially outperforming all prior results and demonstrating the superior quality of the synthetic data for training deep high-performance SNNs.

as dynamic vision sensors (DVS). Unlike conventional cameras that record dense frames at fixed intervals, DVS cameras report pixel-level brightness changes as they occur (Wu et al. 2024). However, this reliance on specialized hardware has created a fundamental data bottleneck that severely impedes the development and adoption of SNNs. The acquisition of large-scale event datasets is a resource-intensive and time-consuming process, resulting in benchmarks that are limited in scale. Furthermore, the quality of existing datasets can be compromised by capture artifacts, such as monitor flicker (Serrano-Gotarredona and Linares-Barranco 2015). The combination of data scarcity and inconsistent quality has led to a persistent performance gap. As shown in Table 1, the accuracy of state-of-the-art networks on event-based ImageNet datasets lags significantly behind their ANN counterparts, with an accuracy of over 70%, casting doubt on their readiness for complex, real-world applications.

To circumvent this data limitation, a common practice involves repeatedly presenting the same image to an SNN

*Corresponding author.

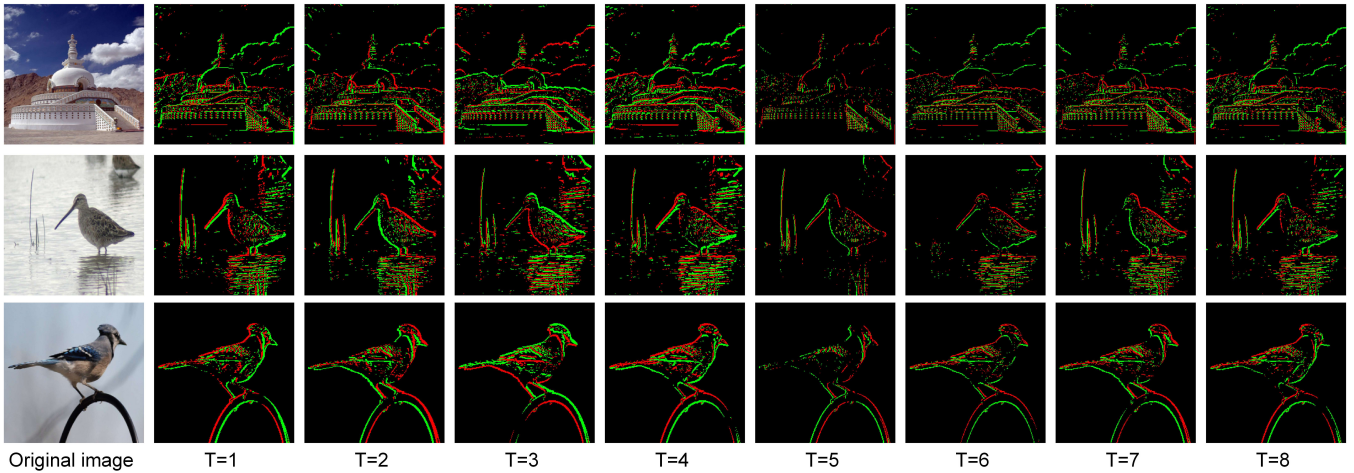


Figure 1: The I2E image-to-event conversion process. A single static RGB image is transformed into an eight-timestep event stream by simulating microsaccadic eye movements. The process effectively captures fine-grained details and salient object contours, producing a sparse data format well-suited for efficient, event-driven processing by SNNs.

at each timestep (Deng et al. 2022; Zhou et al. 2023; Meng et al. 2023; Jiang et al. 2024; Yao et al. 2024). This approach, however, represents a significant compromise, forcing dense, redundant computations that undermine the event-driven paradigm, negating the very energy and latency advantages that make SNNs a compelling alternative to ANNs. This has created an intractable dilemma for the field: either rely on scarce, low-performing real event data or abandon the core principles of neuromorphic computing.

To resolve this challenge, this paper introduces I2E, an ultra-efficient algorithmic framework that converts the vast repository of static images into high-quality event streams in real-time. I2E bridges the gap between large-scale image datasets and the data requirements of high-performance SNNs, enabling them to be trained at scale without compromising their fundamental operating principles. The primary contributions of this work are threefold:

- An algorithmic framework for real-time image-to-event conversion is presented. Its processing speed is over 300x faster than prior methods and up to 30,000x faster than physical acquisition, which for the first time enables the use of on-the-fly data augmentation for SNN training.
- Large-scale, high-quality event-stream datasets, I2E-ImageNet and I2E-CIFAR, are generated. An SNN trained on I2E-ImageNet achieves 60.50% accuracy, establishing a new state-of-the-art for event-based ImageNet and significantly closing the performance gap.
- A highly effective sim-to-real training paradigm is established. By pre-training on synthetic I2E-CIFAR10 data and fine-tuning on the real-world CIFAR10-DVS dataset, an unprecedented accuracy of 92.5% is achieved, demonstrating that I2E-generated data serves as a high-fidelity proxy for real sensor data.

The I2E conversion process, illustrated in Figure 1, effectively preserves crucial visual information within a sparse data format. By open-sourcing the algorithm and the accom-

panying datasets, this work provides the research community with an essential toolkit to overcome the long-standing data bottleneck, thereby accelerating the development of practical, high-performance neuromorphic systems.

Related Work

The acquisition of large-scale, high-quality event-stream data remains a primary obstacle to advancing research on SNNs. Existing approaches to data generation can be broadly categorized into hardware-based capture and algorithmic conversion, each presenting significant drawbacks that have constrained progress in the field.

The most direct approach to data generation involves using DVS to capture events. One strategy is to record real-world scenes, which produces data with high temporal fidelity. However, this process is resource-intensive and slow, resulting in datasets that are often limited in scale and scope, such as DVS-Gesture (Amir et al. 2017) and DailyAction (Liu et al. 2021), or others focused on specific object or action categories (Miao et al. 2019; Bi et al. 2019, 2020; Wang et al. 2021; Vasudevan et al. 2022; Dong et al. 2023; Wang et al. 2024; Sironi et al. 2018; Bolten, Pohle-Frohlich, and Tonnie 2021). The limited sample sizes of these datasets are often insufficient for training the deep SNN architectures required for complex recognition tasks.

To address the issue of scale, an alternative hardware-based strategy involves recording a monitor that displays static images, leading to widely used benchmarks like N-MNIST (Orchard et al. 2015), CIFAR10-DVS (Li et al. 2017), and N-ImageNet (Kim et al. 2021). Although this method increases the number of available samples, it introduces distinct challenges. The resulting datasets may suffer from significant data degradation due to capture artifacts, such as LCD screen flicker. Moreover, the acquisition process remains exceedingly slow. Generating the N-ImageNet dataset, for instance, required several days of continuous recording, rendering the process impractical for expansion

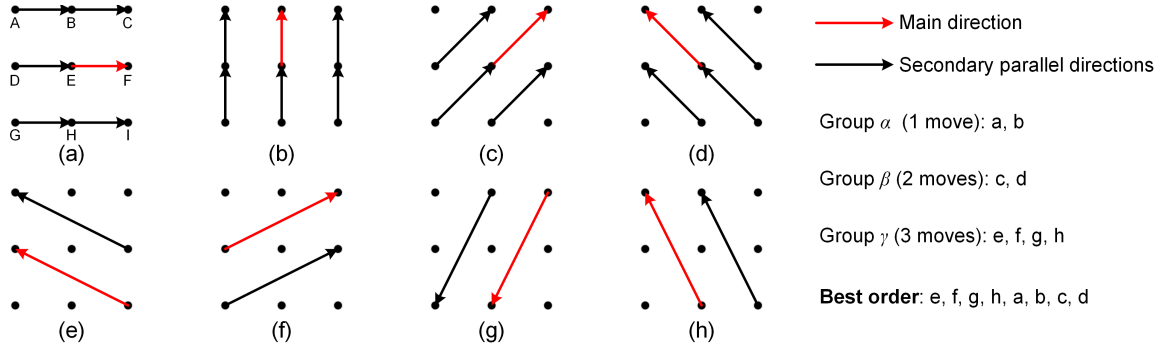


Figure 2: The I2E algorithm simulates microsaccadic eye movements using a source image (point E) and its eight one-pixel-shifted versions (the other points), represented by a 3×3 grid. The intensity change ΔV is calculated by differencing pairs of these images. As shown by the arrows, these differences are classified into eight directional groups, each of which generates the data for one of the eight timesteps in the final event stream.

or modification. A further complication of both hardware-based approaches is the ultra-high temporal resolution of the raw data, which often must be integrated into a smaller number of timesteps for practical use in SNNs (Fang et al. 2021b). This integration process can compromise the sparse, binary nature of the event stream.

To bypass the limitations of physical hardware, several algorithmic methods have been developed to convert conventional visual data into event streams. These include techniques for converting video (Bi and Andreopoulos 2017; Gehrig et al. 2020; Hu, Liu, and Delbruck 2021) and, more recently, static images. A notable example is the ODG algorithm used to create the ES-ImageNet dataset (Lin et al. 2021). While algorithmic conversion drastically reduces cost, existing methods are critically hampered by a computational bottleneck. The generation of a large-scale dataset like ImageNet can take over ten hours on modern hardware. This severe latency makes these algorithms unsuitable for real-time applications and, crucially, precludes the use of on-the-fly data augmentation, a standard and vital technique for training state-of-the-art neural networks.

The I2E framework is designed to resolve these trade-offs. It sidesteps the speed, cost, and quality limitations of hardware acquisition while overcoming the computational bottleneck of prior algorithmic methods, providing a scalable, cost-effective, and practical foundation for training high-performance SNNs within modern deep learning workflows.

Method

This section details the I2E algorithm, a framework for generating high-fidelity event streams from static images in real time. The exposition first presents the core mechanics of the conversion pipeline, followed by a theoretical analysis that quantifies the algorithm’s advantages in terms of speed, energy cost, and information compression.

The I2E Conversion Pipeline

The I2E algorithm (pseudocode see Algorithm 1) transforms a static image into a temporally dynamic event stream through three key stages. The entire pipeline is designed as a

Algorithm 1: The I2E Conversion Algorithm

Input: A batch of RGB images $I \in \mathbb{R}^{B \times 3 \times H \times W}$.
Output: A batch of binary spikes $S \in \mathbb{B}^{T \times B \times 2 \times H \times W}$.

```

1: // Define 8 kernels for 8 motion directions (timesteps)
2:  $v \leftarrow [[9, 4], [4, 3], [3, 8], [8, 1], [5, 6], [5, 2], [5, 3], [5, 1]]$ 
3:  $K \leftarrow \text{zeros}(8, 1, 3, 3)$ 
4: for  $t \in [0, 7]$  do
5:    $y_0, x_0 \leftarrow (v[t][0] - 1) // 3, (v[t][0] - 1) \% 3$ 
6:    $y_1, x_1 \leftarrow (v[t][1] - 1) // 3, (v[t][1] - 1) \% 3$ 
7:    $K[t, 0, y_0, x_0] \leftarrow -1; K[t, 0, y_1, x_1] \leftarrow 1$ 
8: end for
9: // Convert RGB to intensity and compute changes
10:  $V \leftarrow \max(I)$ 
11:  $\Delta V \leftarrow \text{conv2d}(\text{pad}(V, 1), K)$ 
12: // Apply dynamic threshold for event generation
13:  $V_{\text{range}} \leftarrow \max(V) - \min(V)$ 
14:  $S_{th} \leftarrow S_{th0} \cdot V_{\text{range}}$ 
15:  $S_{ON} \leftarrow (\Delta V > S_{th}).\text{float}()$  # ON events
16:  $S_{OFF} \leftarrow (-\Delta V > S_{th}).\text{float}()$  # OFF events
17:  $S \leftarrow \text{stack}([S_{ON}, S_{OFF}], \text{dim} = 2)$ 
18: return  $S.\text{permute}(1, 0, 2, 3, 4)$ 

```

sequence of highly parallelizable tensor operations, making it ideally suited for GPU acceleration.

Stage 1: Intensity Map Generation A DVS camera responds to changes in logarithmic brightness. To efficiently emulate this, a standard RGB image $I_{RGB} \in \mathbb{R}^{3 \times H \times W}$ is first converted into a single-channel intensity map $V \in \mathbb{R}^{1 \times H \times W}$. For this purpose, the Value (V) channel from the HSV color space is used, as it represents the maximum intensity across the R, G, and B channels and can be extracted with negligible computational cost according to Equation 1.

$$V(x, y) = \max(I_R(x, y), I_G(x, y), I_B(x, y)) \quad (1)$$

This choice prioritizes speed while producing an intensity representation analogous to the information captured by a sensor’s photoreceptors. The performance impact of color-to-grayscale conversion is quantified in the ablation study.

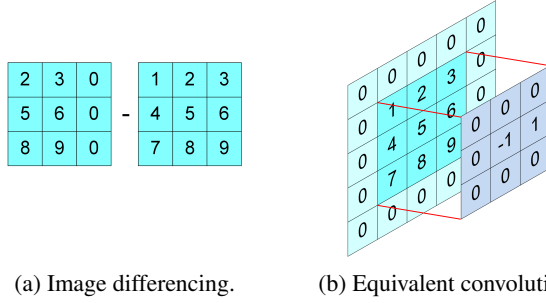


Figure 3: The subtraction of two shifted images is computationally equivalent to a 2D convolution with a sparse kernel.

Stage 2: Event Generation via Spatio-Temporal Convolution A central innovation of I2E is its method for generating temporal dynamics from the static intensity map. The algorithm simulates the effect of microsaccadic eye movements, small, involuntary saccades, by calculating the difference between slightly shifted versions of the intensity map. A naive implementation would require multiple memory-intensive image translation and subtraction operations. Instead, I2E implements this process as a single, highly efficient 2D convolution, as illustrated in Figure 3.

Translating the image by a single pixel in any of the eight directions produces a set of nine images (including the original). The various possible one-pixel motion vectors are classified into eight directional groups, as shown in Figure 2. The difference between pairs of these images simulates the intensity change ΔV that a DVS would capture over a short time interval. For each timestep, a unique 3×3 kernel K_t is constructed. Each kernel is extremely sparse, containing only a single -1 and a single +1 at positions corresponding to the start and end points of the simulated motion vector. The full set of eight intensity-change maps ΔV is then generated in a single, parallel operation, as Equation 2.

$$\Delta V_t = V * K_t \quad (2)$$

This formulation is critical for the algorithm’s real-time performance on GPUs. Ablation study reveals an optimal processing sequence for these groups.

To enhance robustness, a stochastic augmentation strategy is employed during training. For each of the eight directions, a set of equivalent one-pixel shift vectors is defined. During training, one vector is randomly selected from its set to construct the kernel K_t , introducing diversity with no additional computational cost. For inference, a fixed, canonical vector from each set is used to ensure deterministic output.

Stage 3: Adaptive Event Firing The final stage converts the continuous intensity-change maps ΔV into binary spike events. A pixel at position (x, y) fires an *ON* event if $\Delta V(t, x, y)$ exceeds a positive threshold and an *OFF* event if it falls below a negative threshold, as shown in Equation 3, where $S \in \mathbb{B}^{8 \times 2 \times H \times W}$.

$$\begin{cases} S_{(t,0,x,y)} = 1, & \Delta V_{(t,x,y)} > S_{th} \\ S_{(t,1,x,y)} = 1, & \Delta V_{(t,x,y)} < -S_{th} \\ S_{(t,p,x,y)} = 0, & \text{otherwise.} \end{cases} \quad (3)$$

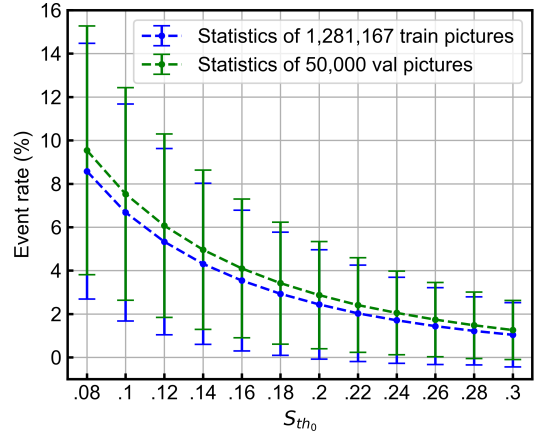


Figure 4: Event rate statistics on ImageNet. $S_{th_0} = 0.12$ is selected to achieve a mean event rate of approximately 5%.

A fixed, global threshold is suboptimal, as it produces inconsistent event rates for images with varying brightness. Therefore, I2E employs a dynamic threshold S_{th} that adapts to each image’s content as shown in Equation 4.

$$S_{th} = S_{th_0} \cdot (\max(V) - \min(V)) \quad (4)$$

where S_{th_0} is a single global sensitivity hyperparameter. This adaptive mechanism ensures a more consistent event sparsity across the dataset, which is critical for robust SNN training. The parameter S_{th_0} directly controls the overall event rate. Figure 4 presents the resulting event rate statistics across the ImageNet dataset for a range of S_{th_0} values. To balance information preservation and computational efficiency, S_{th_0} is selected to achieve a specific target event rate (Lin et al. 2021). For ImageNet, $S_{th_0} = 0.12$ is used to achieve a target event rate of approximately 5%, while for CIFAR datasets, S_{th_0} is set to 0.07.

Efficiency and Information Analysis

The I2E’s design yields significant, quantifiable advantages in computational efficiency and information compression.

Computational, Energy, and Storage Efficiency The convolution-based design of I2E enables unprecedented conversion speed. As shown in Table 2, I2E processes an image on a modern GPU in approximately 0.1 ms. This is orders of magnitude faster than both hardware-based acquisition (e.g., $>30,000\times$ faster than typical DVS camera capture) and prior algorithmic methods like ODG ($>300\times$ faster), which require a day to process the full ImageNet dataset. This real-time capability is a critical advance, enabling the seamless integration of I2E into modern training pipelines that rely on on-the-fly augmentation.

This efficiency translates to substantial energy savings. The energy for a standard ANN convolution is dominated by multiply-accumulate (MAC) operations, $E_{ANN} = N_{ops} \cdot E_{MAC}$. For a ResNet-style first layer, assuming a 45nm process where a 32-bit floating-point MAC costs $E_{MAC} = 4.6$ pJ (Horowitz 2014), the energy consumption is approximately 543 μ J. In contrast, the I2E encoding itself is highly

Dataset	Generation speed (ms/sample)	Resolution	# of samples	# of classes
N-CARS	100	80×40	24,029	2
Poker-DVS	-	32×32	131	4
Bullying10K	2,000 - 20,000	346×260	10,000	10
DVS-Gesture	6,000	128×128	1,342	11
DVS-OUTLAB	500	768×512	47,000	11
DailyDVS-200	1,000 - 13,000	320×240	22,046	200
N-MNIST	300	28×28	60,000 + 10,000	10
MNIST-DVS	2,000 - 4,000	128×128	30,000	10
CIFAR10-DVS	1,200	512×512	10,000	10
DVS-UCF-50	6800	240×180	6,676	50
DVS-Caltech101	300	302×245	8,709	100
N-ImageNet	-	224×224	1,781,167	1,000
N-Omniglot	4,000	346×260	32,460	1,623
ES-ImageNet	29.47	224×224	1,257,035 + 49,881	1,000
I2E-CIFAR10	0.03 (GPU)	128×128	50,000 + 10,000	10
I2E-CIFAR100	0.03 (GPU)	128×128	50,000 + 10,000	100
I2E-ImageNet	0.1 (GPU)	224×224	1,281,167 + 50,000	1,000

Table 2: Comparison of event-based dataset generation speeds. The proposed I2E algorithm demonstrates orders-of-magnitude faster per-sample generation speed compared to both hardware-based acquisition methods and previous algorithmic approaches. Hardware speeds reflect physical capture time, while algorithmic speeds reflect computation time.

efficient ($E_{I2E} \approx 0.36\mu\text{J}$). The SNN layer performs only sparse additions, with energy proportional to the event rate fr and timesteps T , $E_{SNN} = N_{ops} \cdot E_{AC} \cdot T \cdot fr$ and $E_{AC} = 0.9$ pJ. For the I2E-SNN, the cost is approximately $28.68 \mu\text{J}$, representing a $18.9\times$ reduction in first-layer energy consumption compared to the standard ANN approach.

Furthermore, the resulting event-stream data is highly compressible. The I2E-ImageNet dataset, stored as boolean arrays, occupies 47 GB, a 67.8% reduction from the 146 GB of the original JPEG-compressed ImageNet.

Information-Theoretic Analysis To analyze the trade-off between data compression and information preservation, the Shannon entropy of various data representations was computed across the ImageNet dataset. The entropy H of a data source X with discrete symbols x_i and probabilities $p(x_i)$ is given by Equation 5, and measures the average information content per symbol (or pixel).

$$H(X) = - \sum_i p(x_i) \log p(x_i) \quad (5)$$

The original grayscale images and the single-channel Value images contain nearly identical information content, with average entropies of 7.12 ± 0.73 and 7.14 ± 0.76 , respectively. In stark contrast, the final I2E event stream, with a typical event rate of 5%, has a significantly lower entropy of just 1.53 ± 0.60 . This indicates that the I2E conversion achieves substantial information compression, retaining less than 22% of the original data’s entropy. Despite this massive reduction, the empirical performance degradation observed in experiments is comparatively minor. This outcome strongly suggests that the majority of entropy in static im-

ages corresponds to redundant information (such as uniform textures and backgrounds) and validates that the I2E is highly effective at isolating and preserving the sparse, salient features essential for complex recognition tasks.

Experiments

This section empirically validates the effectiveness of the I2E framework. The experiments are designed to assess three key aspects: 1) the performance of SNNs trained on I2E-generated datasets; 2) the transferability of models pre-trained on I2E data to tasks involving real-world neuromorphic sensor data, establishing a new sim-to-real paradigm; and 3) the impact of the algorithm’s specific design choices through a series of ablation studies.

Experimental Setup

Datasets and architectures Performance is evaluated on three standard image recognition benchmarks: CIFAR-10/100 (Krizhevsky, Hinton et al. 2009) and ImageNet (Russakovsky et al. 2015). The corresponding event-based datasets (I2E-CIFAR and I2E-ImageNet) are generated using the I2E algorithm. For these datasets, ImageNet images are resized to 224×224 , while CIFAR images are resized to 128×128 to match the resolution of CIFAR10-DVS. For the sim-to-real evaluation, the CIFAR10-DVS dataset (Li et al. 2017) is used. All experiments employ MS-ResNet architectures (Hu et al. 2024) with LIF neurons (Fang et al. 2021a).

Implementation details Models are trained using the SpikingJelly framework (Fang et al. 2023) with mixed-precision on two NVIDIA RTX 4090 GPUs. The training

Dataset	Architecture	Method	Acc.%
CIFAR10-DVS	ResNet18	baseline	65.6
	ResNet20	baseline	75.56
	ResNet34	transfer	73.72
	SpikingResformer	transfer	84.8
	ResNet18	transfer-I	83.1
	ResNet18	transfer-II	92.5
I2E-CIFAR10	ResNet18	baseline-I	85.07
	ResNet18	baseline-II	89.23
	ResNet18	transfer-I	90.86
I2E-CIFAR100	ResNet18	baseline-I	51.32
	ResNet18	baseline-II	60.68
	ResNet18	transfer-I	64.53

Table 3: Performance on CIFAR datasets. Transfer-I denotes fine-tuning after pre-training on I2E-ImageNet. Transfer-II denotes fine-tuning after pre-training on I2E-CIFAR10. The transfer-II result of 92.5% establishes a new state-of-the-art on real-world CIFAR10-DVS, demonstrating the effectiveness of the proposed sim-to-real training paradigm.

employs a cross-entropy loss with label smoothing ($\epsilon = 0.1$) and the SGD optimizer. Models for CIFAR and ImageNet are trained for 256 and 128 epochs, respectively. The initial learning rate is set to 0.1, with weight decay of $2e-4$ for CIFAR and $1e-5$ for ImageNet.

Data formats The generated I2E datasets are provided in two formats: dense boolean tensors, convenient for direct loading into deep learning frameworks, and sparse coordinate lists, which are highly compressed and suitable for applications that process events individually.

Performance on I2E-Generated Datasets

A key advantage of I2E’s real-time nature is its compatibility with standard on-the-fly data augmentation pipelines, a technique precluded by the static nature of previous event datasets. To quantify this benefit, two baseline conditions were evaluated. In Baseline-I, only minimal augmentation, random horizontal flipping, was used. In Baseline-II, a full suite of standard augmentations such as random cropping was applied to the source images before I2E conversion.

As shown in Table 1 and Table 3, models trained on I2E data achieve state-of-the-art results. On I2E-ImageNet, MS-ResNet34 (Baseline-II) reaches 60.50% accuracy, surpassing the best prior result on other event-based ImageNet datasets by over 8%. The dramatic performance increase from Baseline-I to Baseline-II across all datasets demonstrates that I2E is not only capable of generating high-quality event data but also uniquely enables the modern training strategies required to unlock the full potential of deep SNNs.

Transfer Learning: A New Paradigm for SNNs

The most significant contribution of this work is the establishment of a practical and effective pre-training paradigm for SNNs. By providing a virtually unlimited source of

Method	Dynamic Threshold	Random Selection	Random Crop	Acc. %
ablation-1	×	×	×	47.22
baseline-I	✓	×	×	48.30
ablation-2	✓	✓	×	49.01
baseline-II	✓	✓	✓	57.97

Table 4: Ablation study of I2E components on I2E-ImageNet with ResNet18. Dynamic thresholding, random selection, and compatibility with standard augmentations are all essential for achieving optimal performance.

Dataset	Acc.%					
	$\alpha\beta\gamma$	$\alpha\gamma\beta$	$\beta\alpha\gamma$	$\beta\gamma\alpha$	$\gamma\alpha\beta$	$\gamma\beta\alpha$
CIFAR10	87.96	88.94	87.36	88.88	89.23	88.60
CIFAR100	56.10	59.43	55.11	59.25	60.68	60.12

Table 5: Ablation on the timestep processing order. Presenting groups with higher event rates first (the $\gamma\alpha\beta$ sequence) consistently yields the best performance.

low-cost, high-quality synthetic event data, I2E enables robust pre-training for subsequent fine-tuning on smaller, real-world event datasets, thereby addressing the critical data scarcity problem in neuromorphic engineering.

Transferability across I2E datasets First, the effectiveness of transfer learning within the I2E ecosystem was established. As shown in Table 3, a model pre-trained on the large-scale I2E-ImageNet dataset and then fine-tuned on I2E-CIFAR demonstrates significant performance gains. Accuracy on I2E-CIFAR10 improves from 89.23% to 90.86%, while on the more challenging I2E-CIFAR100, accuracy sees a substantial boost from 60.68% to 64.53%. This confirms that features learned on I2E-ImageNet are general and transferable to other I2E-generated tasks.

Bridging the sim-to-real gap The key experiment involves transferring knowledge from I2E-generated data to a real-world DVS sensor dataset. A model was pre-trained on synthetic I2E-CIFAR10 data and then fine-tuned on the real-world CIFAR10-DVS dataset. The result, as shown in Table 3, achieves a new state-of-the-art accuracy of 92.5%, outperforming the previous best by a remarkable 7.7%. This successful sim-to-real transfer is a crucial finding, as it demonstrates that the event streams produced by I2E serve as a high-fidelity proxy for real sensor data. It validates a powerful new workflow for the field: leverage vast static image libraries to pre-train robust SNNs via I2E, and then fine-tune them with a limited amount of costly, real-world DVS data to achieve state-of-the-art performance. This paradigm mitigates the data acquisition bottleneck that has long hindered progress in neuromorphic computing.

Method	Channel	Timestep	Acc.%
RGB	3	4	65.68
V	1	4	62.21
Event	2	8	59.28

Table 6: Analysis of performance impact from RGB-to-Value conversion on ImageNet.

Ablation Studies and Analysis

To validate the design choices of the I2E algorithm, a series of ablation studies and analyses were conducted.

Impact of algorithmic components The ablation study on ImageNet, as shown in Table 4, confirms the importance of I2E’s core components by building the algorithm from the ground up. Starting with a naive conversion (fixed threshold), the model achieves only 47.22% accuracy. Introducing the dynamic threshold stabilizes the event rate and improves performance to 48.30%. Adding the random selection provides essential data augmentation, further boosting accuracy to 49.01%. Finally, enabling standard augmentations such as random cropping, which is only possible due to I2E’s real-time nature, provides the largest benefit, increasing accuracy to 57.97%. This highlights the synergy between the algorithm’s design and modern training practices.

Analysis of timestep order The sequence in which event frames are processed affects performance. The eight motion vectors were categorized into three groups (α, β, γ) based on the magnitude of their modulus, as Figure 2, which correlates with the resulting event rate. As shown in Table 5, ordering the groups to present frames with higher event rates first (the $\gamma\alpha\beta$ sequence) consistently yields the best performance on both CIFAR-10 and CIFAR-100. Corresponding to the eight timesteps, the best order is: e, f, g, h, a, b, c, d.

Analysis of conversion loss and timesteps An analysis was conducted to quantify information loss during conversion. As shown in Table 6, converting an RGB image to a single-channel Value map results in a performance drop from 65.68% to 62.21%, defining a practical upper bound for the event-based approach. The final I2E-trained model achieves 59.28%, indicating that while the conversion is highly effective, the inherent sparsity of events still introduces a minor performance trade-off. Furthermore, the number of timesteps can be adjusted to balance accuracy and data compression as shown in Figure 5. Reducing the timesteps to just 2 still yields an accuracy of 51.97% on ImageNet, which is competitive with prior work, while increasing the data compression ratio to 91.95%.

Discussion and Limitations

The primary implication of the I2E is the establishment of a practical and highly effective pre-train, then fine-tune workflow for the neuromorphic domain. This approach directly addresses the data scarcity and quality bottleneck that has long constrained SNN research. The successful sim-to-real

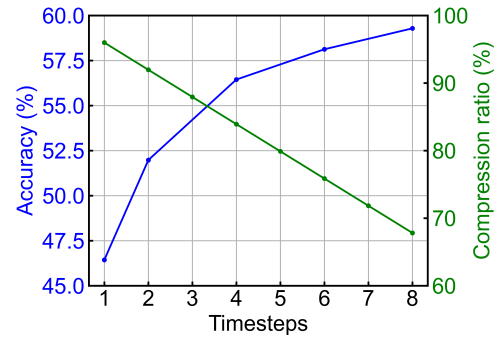


Figure 5: Trade-off between timesteps, accuracy, and data compression on ImageNet. Using more timesteps improves accuracy at the cost of a lower data compression ratio.

transfer experiment, in which a model pre-trained on synthetic I2E data achieved state-of-the-art performance on the real-world CIFAR10-DVS dataset, strongly validates this paradigm. It demonstrates that I2E-generated data serves as a high-fidelity proxy for physical sensor data, effectively decoupling SNN model development from the slow and costly process of hardware-based data acquisition.

This development lowers the barrier to entry for SNN research and development. By enabling ubiquitous, low-cost RGB cameras to function as effective event-based sensors through a software layer, I2E makes the design of energy-efficient SNNs more economically and logistically viable. Furthermore, the algorithm’s real-time nature unlocks new research avenues. The systematic exploration of data augmentation strategies for event streams, a critical area for improving generalization, was previously inaccessible due to the static nature of existing event datasets. While this study focused on validating the I2E paradigm for classification, extending this pre-training workflow to other complex tasks, such as detection and segmentation, or other event-driven tasks, remains a key direction for future work.

Conclusion

This work introduced I2E, an algorithmic framework that resolves a critical data bottleneck for SNNs by converting static images into high-fidelity event streams in real-time. The method’s efficiency, which is orders of magnitude faster than prior approaches, uniquely enables the use of modern on-the-fly data augmentation pipelines for SNN training. The quality of the generated data was demonstrated by training a deep SNN on the new I2E-ImageNet dataset to a state-of-the-art accuracy of 60.50%. Critically, this work established a powerful sim-to-real paradigm by pre-training a model on synthetic I2E data and fine-tuning it on the real-world CIFAR10-DVS dataset, achieving an unprecedented accuracy of 92.5%. By open-sourcing the algorithm and datasets, this research provides the community with an essential toolkit to bridge the data gap, accelerating the development of high-performance, practical neuromorphic systems. This work thus paves the way for deploying SNNs in complex, real-world applications where both high performance and extreme energy efficiency are required.

Acknowledgments

This work was supported in part by the Scientific and Technological Innovation (STI) 2030-Major Projects under Grant 2022ZD0209700, in part by Sichuan Science and Technology Program under Grant 2024ZDZX0001 and Grant 2024ZYD0253, and in part by Shenzhen Natural Science Foundation under JCYJ20250604180428038.

References

- Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Nayak, T.; Andreopoulos, A.; Garreau, G.; Mendoza, M.; et al. 2017. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7243–7252.
- Bi, Y.; and Andreopoulos, Y. 2017. PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams. In *2017 IEEE International Conference on Image Processing (ICIP)*, 1990–1994. IEEE.
- Bi, Y.; Chadha, A.; Abbas, A.; Bourtsoulatze, E.; and Andreopoulos, Y. 2019. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF international conference on computer vision*, 491–501.
- Bi, Y.; Chadha, A.; Abbas, A.; Bourtsoulatze, E.; and Andreopoulos, Y. 2020. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 29: 9084–9098.
- Bolten, T.; Pohle-Frohlich, R.; and Tonnies, K. D. 2021. Dvs-outlab: A neuromorphic event-based long time monitoring dataset for real-world outdoor scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1348–1357.
- Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.
- Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal Efficient Training of Spiking Neural Network via Gradient Reweighting. In *International Conference on Learning Representations*.
- Dong, Y.; Li, Y.; Zhao, D.; Shen, G.; and Zeng, Y. 2023. Bullying10k: a large-scale neuromorphic dataset towards privacy-preserving bullying recognition. *Advances in Neural Information Processing Systems*, 36: 1923–1937.
- Fang, W.; Chen, Y.; Ding, J.; Yu, Z.; Masquelier, T.; Chen, D.; Huang, L.; Zhou, H.; Li, G.; and Tian, Y. 2023. Spiking-jelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40): eadi1480.
- Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021a. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 21056–21069.
- Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; and Tian, Y. 2021b. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2661–2671.
- Gehrig, D.; Gehrig, M.; Hidalgo-Carrió, J.; and Scaramuzza, D. 2020. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3586–3595.
- Horowitz, M. 2014. 1.1 computing’s energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, 10–14. IEEE.
- Hu, Y.; Deng, L.; Wu, Y.; Yao, M.; and Li, G. 2024. Advancing spiking neural networks toward deep residual learning. *IEEE transactions on neural networks and learning systems*, 36(2): 2353–2367.
- Hu, Y.; Liu, S.-C.; and Delbruck, T. 2021. v2e: From video frames to realistic DVS events. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1312–1321.
- Jiang, H.; Zoonekynd, V.; De Masi, G.; Gu, B.; and Xiong, H. 2024. TAB: Temporal Accumulated Batch normalization in spiking neural networks. In *The Twelfth International Conference on Learning Representations*.
- Kim, J.; Bae, J.; Park, G.; Zhang, D.; and Kim, Y. M. 2021. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2146–2156.
- Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.
- Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11: 309.
- Lin, Y.; Ding, W.; Qiang, S.; Deng, L.; and Li, G. 2021. Es-imagenet: A million event-stream classification dataset for spiking neural networks. *Frontiers in neuroscience*, 15: 726582.
- Liu, Q.; Xing, D.; Tang, H.; Ma, D.; and Pan, G. 2021. Event-based Action Recognition Using Motion Information and Spiking Neural Networks. In *IJCAI*, 1743–1749.
- Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z.-Q. 2023. Towards memory-and time-efficient backpropagation for training spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6166–6176.
- Merolla, P. A.; Arthur, J. V.; Alvarez-Icaza, R.; Cassidy, A. S.; Sawada, J.; Akopyan, F.; Jackson, B. L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197): 668–673.
- Miao, S.; Chen, G.; Ning, X.; Zi, Y.; Ren, K.; Bing, Z.; and Knoll, A. 2019. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. *Frontiers in neurorobotics*, 13: 38.

- Orchard, G.; Jayawant, A.; Cohen, G. K.; and Thakor, N. 2015. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9: 437.
- Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W.; et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, 572(7767): 106–111.
- Roy, K.; Jaiswal, A.; and Panda, P. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784): 607–617.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.
- Serrano-Gotarredona, T.; and Linares-Barranco, B. 2015. Poker-DVS and MNIST-DVS. Their history, how they were made, and other details. *Frontiers in neuroscience*, 9: 481.
- Sironi, A.; Brambilla, M.; Bourdis, N.; Lagorce, X.; and Benosman, R. 2018. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1731–1740.
- Subbulakshmi Radhakrishnan, S.; Sebastian, A.; Oberoi, A.; Das, S.; and Das, S. 2021. A biomimetic neural encoder for spiking neural network. *Nature communications*, 12(1): 2143.
- Vasudevan, A.; Negri, P.; Di Ielsi, C.; Linares-Barranco, B.; and Serrano-Gotarredona, T. 2022. SL-Animals-DVS: event-driven sign language animals dataset. *Pattern Analysis and Applications*, 1–16.
- Wang, Q.; Xu, Z.; Lin, Y.; Ye, J.; Li, H.; Zhu, G.; Ali Shah, S. A.; Bennamoun, M.; and Zhang, L. 2024. Dailydvs-200: A comprehensive benchmark dataset for event-based action recognition. In *European Conference on Computer Vision*, 55–72. Springer.
- Wang, Y.; Zhang, X.; Shen, Y.; Du, B.; Zhao, G.; Cui, L.; and Wen, H. 2021. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3436–3449.
- Wu, W.-Q.; Wang, C.-F.; Han, S.-T.; and Pan, C.-F. 2024. Recent advances in imaging devices: Image sensors and neuromorphic vision sensors. *Rare Metals*, 43(11): 5487–5515.
- Xu, Q.; Qi, Y.; Yu, H.; Shen, J.; Tang, H.; Pan, G.; et al. 2018. Csnns: an augmented spiking based framework with perceptron-inception. In *IJCAI*, volume 1646.
- Yao, M.; Hu, J.; Hu, T.; Xu, Y.; Zhou, Z.; Tian, Y.; XU, B.; and Li, G. 2024. Spike-driven Transformer V2: Meta Spiking Neural Network Architecture Inspiring the Design of Next-generation Neuromorphic Chips. In *The Twelfth International Conference on Learning Representations*.
- Zenke, F.; Bohtë, S. M.; Clopath, C.; Comşa, I. M.; Gölitz, J.; Maass, W.; Masquelier, T.; Naud, R.; Neftci, E. O.; Petrovici, M. A.; et al. 2021. Visualizing a joint future of neuroscience and neuromorphic engineering. *Neuron*, 109(4): 571–575.
- Zhang, Y.; Qu, P.; Ji, Y.; Zhang, W.; Gao, G.; Wang, G.; Song, S.; Li, G.; Chen, W.; Zheng, W.; et al. 2020. A system hierarchy for brain-inspired computing. *Nature*, 586(7829): 378–384.
- Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; YAN, S.; Tian, Y.; and Yuan, L. 2023. Spikformer: When Spiking Neural Network Meets Transformer. In *The Eleventh International Conference on Learning Representations*.

Spiking Neural Networks

The fundamental computational unit of a spiking neural network (SNN) is the spiking neuron. The dynamics of a spiking neuron are governed by a discrete-time process that unfolds in three distinct stages: membrane potential integration, spike generation, and potential post-spike reset. This process can be formulated as follows:

$$H[t] = f(V[t-1], X[t]), \quad (6)$$

$$S[t] = \Theta(H[t] - V_{th}), \quad (7)$$

$$V[t] = H[t](1 - S[t]) + V_{reset}S[t], \quad (8)$$

Here, at each timestep t , $X[t]$ represents the input current to the neuron. The function $f(\cdot)$ in Equation 6 integrates this input with the previous state's membrane potential $V[t-1]$, to produce an intermediate potential $H[t]$. The neuron generates a binary output spike, $S[t] \in \{0, 1\}$, if this potential $H[t]$ exceeds a predefined firing threshold V_{th} , which is set to 1. This firing mechanism is modeled by the Heaviside step function $\Theta(\cdot)$, as shown in Equation 7, where $\Theta(x) = 1$ for $x > 0$ and is 0 otherwise. Following a spike ($S[t] = 1$), the neuron's membrane potential $V[t]$ is reset to $V_{reset} = 0$. Otherwise, it retains the value of $H[t]$, as Equation 8.

The specific computational behavior of the neuron is determined by the choice of the integration function $f(\cdot)$. The integration dynamics of the widely used Leaky Integrate-and-Fire (LIF) neuron are given by:

$$H[t]_{LIF} = V[t-1] - \frac{1}{\tau}(V[t-1] - V_{reset}) + X[t] \quad (9)$$

where τ is the membrane time constant, which controls the rate of potential decay. We set $\tau = 2$ in our experiments. For constructing deep residual networks, we adopt the Membrane-Shortcut ResNet (MS-ResNet) architecture, where the residual connection is applied to the membrane potential $V[t]$ rather than the sparse spike signal $S[t]$.

A significant challenge in training SNNs is the non-differentiable nature of the Heaviside function used for spike generation. To enable gradient-based optimization, we employ the surrogate gradient method. This approach replaces the true gradient of $\Theta(\cdot)$ during the backward pass with the gradient of a smooth, differentiable surrogate function. Specifically, we use an arctan-based surrogate, defined as:

$$g(x) = \frac{1}{\pi} \arctan\left(\frac{\pi}{2}\alpha x\right) + \frac{1}{2} \quad (10)$$

Its derivative, which is used in the backward pass, is:

$$g'(x) = \frac{2\alpha}{4 + (\pi\alpha x)^2} \quad (11)$$

The hyperparameter α controls the steepness of the surrogate function, effectively modulating the sharpness of the approximated gradient. We set $\alpha = 2$. A visual comparison of the Heaviside step function, its surrogate, and the surrogate's gradient is provided in Figure 6.

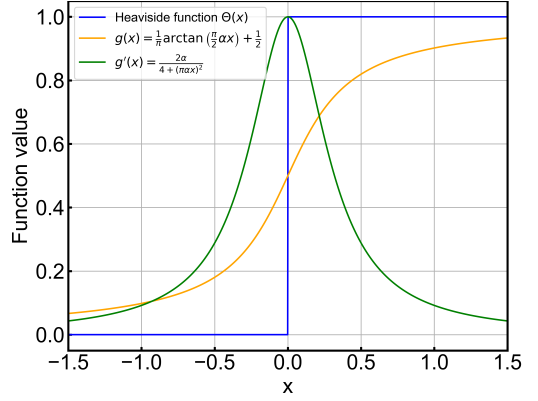


Figure 6: Visualization of the surrogate gradient method. The blue line represents the Heaviside step function used in the forward pass. The orange line shows the smooth arctan-based surrogate function $g(x)$. The green line illustrates the derivative of the surrogate function $g'(x)$, which is used to compute gradients during the backward pass.

Detailed Energy Consumption Analysis

This section provides a detailed breakdown of the theoretical energy consumption calculations referenced in the main text. The fundamental energy-consuming operation in a conventional convolutional layer is the multiply-accumulate (MAC) operation, whereas in a spiking convolutional layer, it is the accumulate (AC) operation, triggered only upon receiving an input spike. Following the prior work, we adopt energy cost values of $E_{mac} = 4.6$ pJ for a 32-bit floating-point MAC operation and $E_{ac} = 0.9$ pJ for an AC operation.

The energy consumption for an ANN layer (E_{ANN}) and an SNN layer (E_{SNN}) can thus be modeled as:

$$E_{ANN} = N_{ops} \cdot E_{mac} \quad (12)$$

$$E_{SNN} = N_{ops} \cdot fr \cdot T \cdot E_{ac} \quad (13)$$

where N_{ops} is the total number of synaptic operations (MAC or AC) in the layer, fr is the average firing rate of the neurons, and T is the number of simulation timesteps. The theoretical number of operations for a standard convolutional layer is calculated as:

$$N_{ops} = K^2 \cdot C_{in} \cdot C_{out} \cdot H_{out} \cdot W_{out} \quad (14)$$

Here, K is the kernel size, C_{in} and C_{out} are the input and output channel counts, and H_{out} and W_{out} are the spatial dimensions of the output feature map.

To illustrate the practical implications, we analyze the first layer of a ResNet-style architecture processing inputs from the ImageNet dataset ($224 \times 224 \times 3$). This layer has a kernel size $K = 7$, stride 2, $C_{in} = 3$, $C_{out} = 64$, and produces an output of size 112×112 .

For the conventional ANN, the energy cost is:

$$E_{ANN} = (7^2 \cdot 3 \cdot 64 \cdot 112^2) \cdot 4.6 \text{ pJ} \approx 543 \mu\text{J} \quad (15)$$

For our proposed I2E-SNN, the first layer processes the output of the I2E module, which has $C_{in} = 2$. Assuming a typical firing rate $fr = 5\%$ and $T = 8$, the energy cost is:

$$E_{SNN} = (7^2 \cdot 2 \cdot 64 \cdot 112^2) \cdot 0.05 \cdot 8 \cdot 0.9 \text{ pJ} \approx 28 \mu\text{J} \quad (16)$$

The I2E encoding itself functions as a lightweight convolutional layer with $C_{\text{in}} = 1$, $C_{\text{out}} = 8$ (for $T = 8$), operating on the full input resolution. Its energy cost is minimal:

$$E_{\text{I2E}} = (1^2 \cdot 1 \cdot 8 \cdot 224^2) \cdot 0.9 \text{ pJ} \approx 0.36 \mu\text{J} \quad (17)$$

The total energy for the first effective layer of our model is $E_{\text{SNN}} + E_{\text{I2E}} \approx 28.68 \mu\text{J}$. Compared to the standard ANN’s first layer, this represents a **18.9×** reduction in energy consumption. This energy efficiency can be further improved, for instance, reducing the timestep to $T = 2$ lowers the first layer’s energy consumption to approximately $7.17 \mu\text{J}$, achieving a **75.7×** reduction relative to the ANN baseline.

Data Representation and Compression

To facilitate diverse experimental setups and hardware targets, we converted the static ImageNet dataset into two distinct event-based formats. This section details the structure of these formats and analyzes the resulting data compression. The original ImageNet dataset, comprising static JPEG images, occupies 146 GB of storage.

Dense Tensor Representation

The first format is a dense tensor representation, designed for seamless integration with deep learning frameworks that expect structured, multi-dimensional arrays. Each image is converted into a Boolean tensor with shape $[T, C, H, W]$, where:

- T is the number of simulation timesteps.
- C represents the polarity channels (e.g., ON/OFF events), resulting in $C = 2$.
- H and W are the spatial dimensions of the image.

For an ImageNet image processed with $T = 8$ timesteps, this results in a tensor of shape $[8, 2, 224, 224]$. Each tensor is stored as a separate *.npz* file. The completely converted dataset in this format, including both training and validation sets, occupies **47 GB**.

Sparse Event-Stream Representation

The second format mimics the native output of DVS. It is a sparse, coordinate-list representation where only pixel changes are recorded. Each image is converted into a list of events, where each event is a tuple (t, p, x, y) representing:

- t : The discrete timestep at which the event occurred.
- p : The polarity of the event (e.g., 0 for OFF, 1 for ON).
- x, y : The spatial coordinates of the pixel that fired.

This list is stored as a multi-column matrix of type *uint8* in an *.npz* file. This event-based format is inherently more efficient for sparse data. The total size of the ImageNet dataset converted to this format is **44 GB**.

Compression Analysis

Both event-based representations offer substantial data compression compared to the original 146 GB dataset. The compression ratio is calculated as $1 - \frac{\text{Encoded Size}}{\text{Original Size}}$.

Dataset	Learning Rate	Weight Decay	Batch Size	Epoch
I2E-CIFAR	0.1	2e-4	128	256
ImageNet	0.1	2e-5	128	128
I2E-ImageNet	0.1	1e-5	128	128
CIFAR10-DVS	0.1	1e-5	32	128

Table 7: Hyperparameters for training models from scratch.

Dataset	Method	Learning Rate	Weight Decay
I2E-CIFAR	transfer-I	0.01	2e-4
I2E-ImageNet	transfer-I	0.05	1e-5
CIFAR10-DVS	transfer-I	0.01	1e-3
	transfer-II	0.001	0

Table 8: Hyperparameters for fine-tuning models.

- For the dense tensor format ($T = 8$):

$$\text{Compression Ratio} = 1 - \frac{47 \text{ GB}}{146 \text{ GB}} \approx 67.81\% \quad (18)$$

- For the sparse event-stream format:

$$\text{Compression Ratio} = 1 - \frac{44 \text{ GB}}{146 \text{ GB}} \approx 69.86\% \quad (19)$$

Furthermore, the size of both formats scales linearly with the number of timesteps. This allows for even greater compression when fewer timesteps are required. For instance, in our ablation study using $T = 2$, the storage requirement for the dense format would be reduced by a factor of four. This yields a significantly higher compression ratio:

$$\text{Compression Ratio} \approx 1 - \frac{47/4 \text{ GB}}{146 \text{ GB}} \approx 91.95\% \quad (20)$$

This highlights the efficiency of event-based encoding, particularly for applications where a short temporal window is sufficient.

Experimental Details

Experimental Setup and Training Configuration

All experiments were conducted on an Ubuntu 22.04 server equipped with two NVIDIA RTX 4090 GPUs. To ensure reproducibility, all random seeds for data shuffling, initialization, and other stochastic processes were fixed to 2024.

We employed the SGD optimizer with a momentum of 0.9 for all training procedures. The learning rate was decayed using a cosine annealing schedule. Following common practice in SNN training to enhance stability and performance, the gradient flow through the reset mechanism (Equation 8) was detached during the backward pass. The specific hyperparameters for each dataset and method are detailed in Table 7 (for training from scratch) and Table 8 (for fine-tuning).

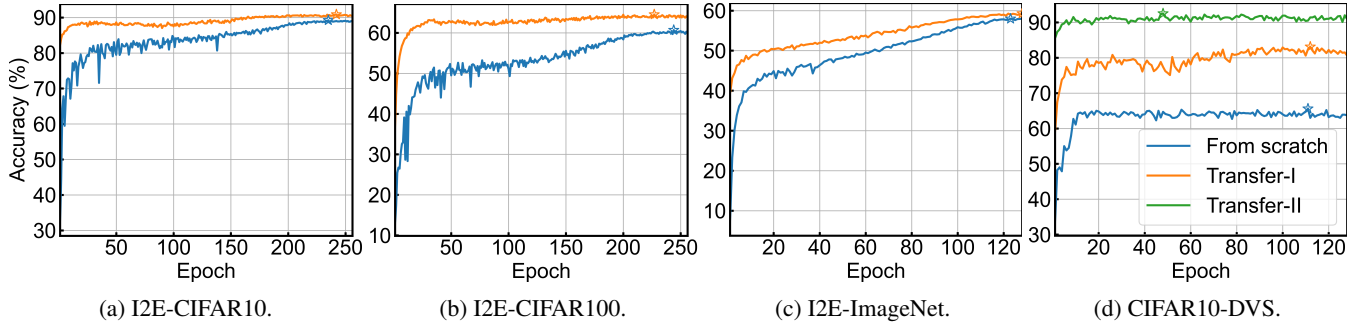


Figure 7: Validation accuracy curves on four datasets. Each plot compares the performance of models trained from scratch against those fine-tuned using transfer learning strategies.

Dataset-Specific Protocols

CIFAR10-DVS This native event-based dataset contains 10,000 samples. We designated 90% of the data for the training set and the remaining 10% for the validation set. The raw event streams were integrated into dense frames over $T = 8$ timesteps. No data augmentation techniques were applied during the training process.

I2E-converted datasets For I2E-CIFAR and I2E-ImageNet, the static images were converted into event-based formats as described in the main text. The training and fine-tuning followed the hyperparameter settings specified in Tables 7 and 8.

Transfer Learning and Fine-Tuning Strategies

We investigated several transfer learning scenarios to evaluate the efficacy of pre-training. The specific fine-tuning hyperparameters are listed in Table 8.

- **I2E-ImageNet:** The model was first pre-trained on the standard static ImageNet dataset. The first convolutional layer’s weights were adapted to the dual-channel event input by removing the parameters corresponding to the third input channel of the original RGB data. The model was then fine-tuned on the I2E-ImageNet dataset.
- **I2E-CIFAR:** The models were pre-trained on the large-scale I2E-ImageNet dataset and subsequently fine-tuned on the I2E-CIFAR datasets.
- **CIFAR10-DVS:** We explored two pre-training strategies:
 - **Transfer-I:** Pre-trained on I2E-ImageNet.
 - **Transfer-II:** Pre-trained on I2E-CIFAR10.

As illustrated by the training accuracy curves in Figure 7, pre-training consistently provided a significant performance improvement across all evaluated datasets.

Ablation Study Details

For our ablation studies, the default simulation timestep was set to $T = 4$ unless specified otherwise.

Timestep reduction In the experiment analyzing the impact of fewer timesteps, we constructed datasets with $T < 8$ by systematically selecting the first n timesteps from the fully generated 8-timestep data.

Group	Time	Group	Time
$\alpha\beta\gamma$	abcdefgh	$\alpha\gamma\beta$	abefghcd
$\beta\alpha\gamma$	cdabefgh	$\beta\gamma\alpha$	cdefghab
$\gamma\alpha\beta$	efghabcd	$\gamma\beta\alpha$	efghcdab

Table 9: Mapping of group orders to temporal sequence.

Temporal order In the experiments studying the influence of temporal sequence, the 8 timesteps were divided into three groups (α, β, γ). Table 9 details the specific mapping between group permutations and the resulting temporal order of the timesteps.

Visualization of the I2E Conversion Process

To illustrate the I2E algorithm’s pipeline, we provide a full conversion example for a sample image in Figure 8 and 9. This process clearly demonstrates the three main stages of converting a static image into a sparse event stream:

- **Stage 1 (Row 1):** The original RGB image (a) is converted into a single-channel intensity V-map (b), which serves as the basis for subsequent calculations.
- **Stage 2 (Row 2):** The V-map is processed with 8 directional convolutional kernels, generating 8 timesteps of intensity change maps (c). The figure shows the float-point values for the two channels (ON/OFF), representing the magnitude and direction of brightness changes.
- **Stage 3 (Row 3):** The float-point values are compared against the adaptive threshold to generate the final binary event stream (d). This is the sparse data fed into the SNN.

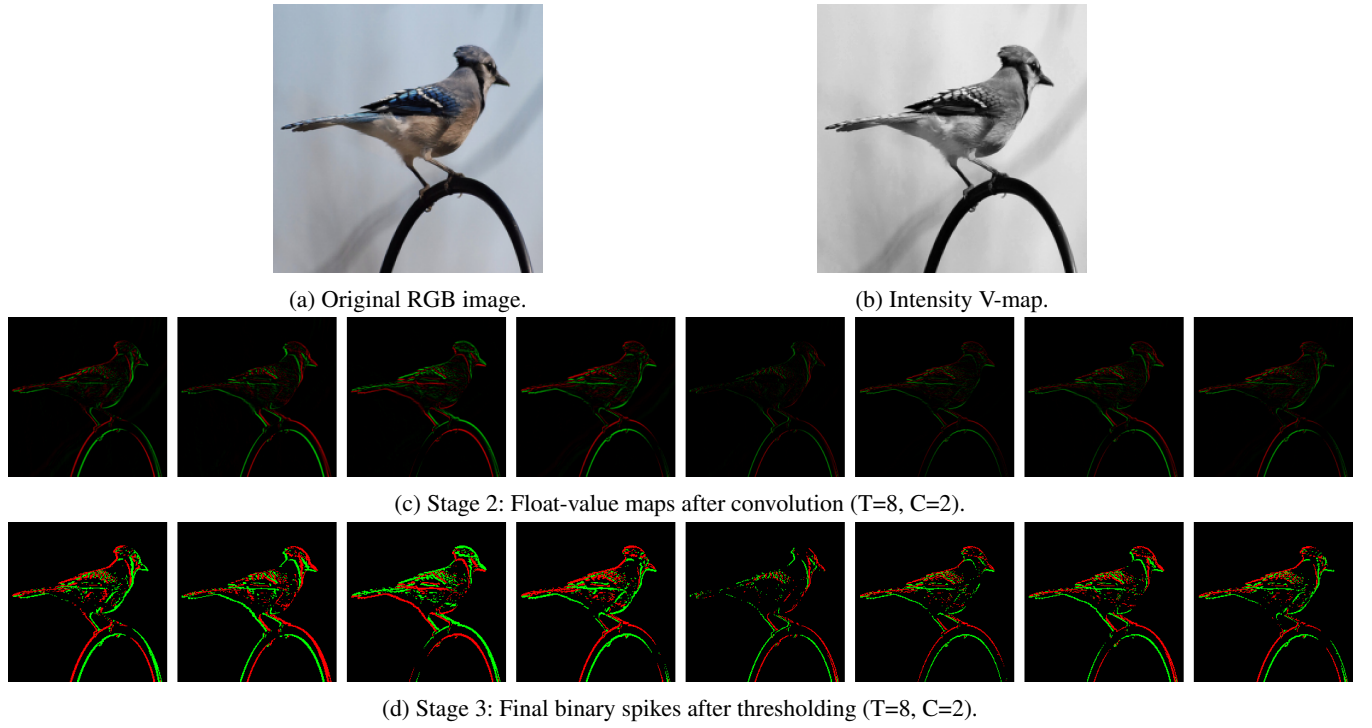


Figure 8: A complete visualization of the I2E conversion process. The figure illustrates the complete data flow for a single sample, progressing from (a) original RGB image to (b) its intensity V-map, then to (c) the intermediate float-value maps, and finally (d) the binary event stream.

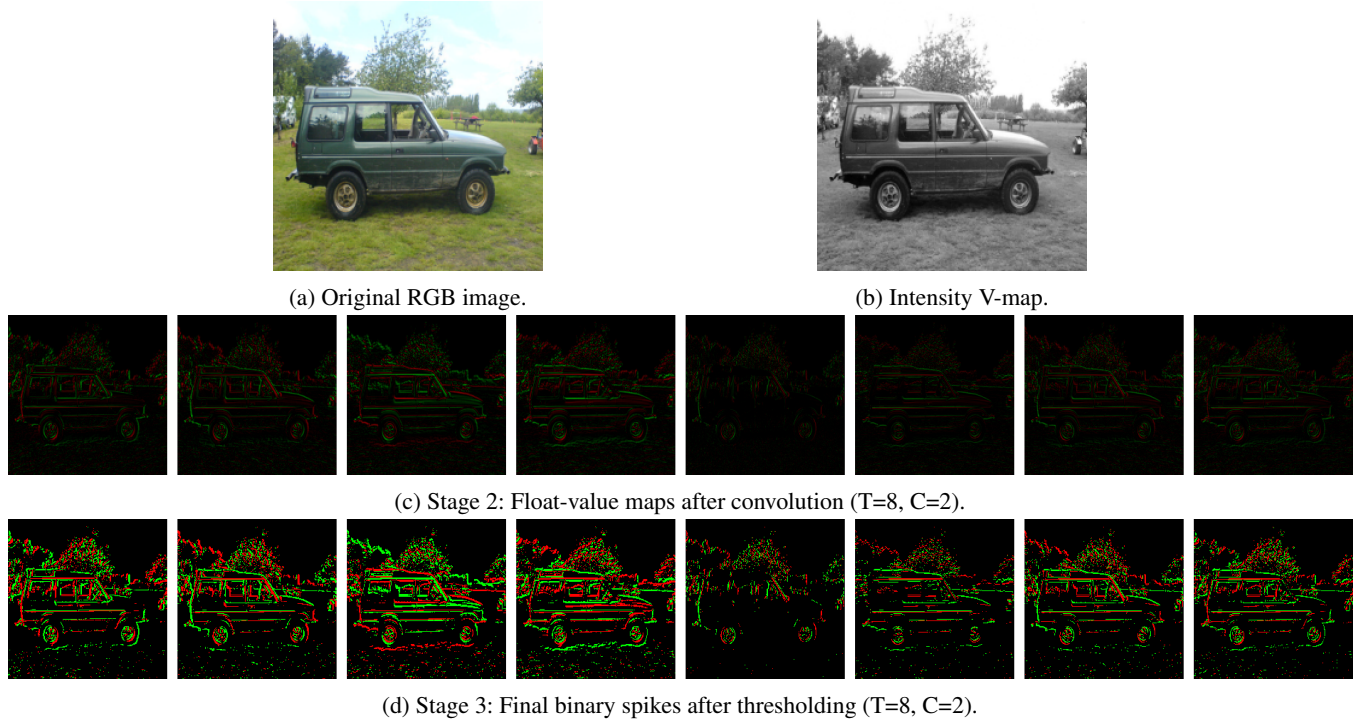


Figure 9: Another example.