HyperE2VID: Improving Event-Based Video Reconstruction via Hypernetworks

Burak Ercan, Onur Eker, Canberk Saglam, Aykut Erdem, *Senior Member, IEEE*, Erkut Erdem *Senior Member, IEEE*

***Abstract*—Event-based cameras are becoming increasingly popular for their ability to capture high-speed motion with low latency and high dynamic range. However, generating videos from events remains challenging due to the highly sparse and varying nature of event data. To address this, in this study, we propose HyperE2VID, a dynamic neural network architecture for event-based video reconstruction. Our approach uses hypernetworks to generate per-pixel adaptive filters guided by a context fusion module that combines information from event voxel grids and previously reconstructed intensity images. We also employ a curriculum learning strategy to train the network more robustly. Our comprehensive experimental evaluations across various benchmark datasets reveal that HyperE2VID not only surpasses current state-of-the-art methods in terms of reconstruction quality but also achieves this with fewer parameters, reduced computational requirements, and accelerated inference times.**

***Index Terms*—Event-based vision, video reconstruction, dynamic neural networks, hypernetworks, dynamic convolutions.**

I. Introduction

In the past decade, the field of computer vision has seen astonishing progress in many different tasks, thanks to modern deep learning methodologies and recent neural architectures. But, despite all these advances, current artificial vision systems still fall short on dealing with some real-world situations involving high-speed motion scenes with high dynamic range, as compared to their biological counterparts. Some of these shortcomings can be attributed to the classical frame-based acquisition and processing pipelines, since the traditional frame-based sensors have some problems such as motion blur and low dynamic range due to the underlying basic principles used for collecting light.

The recently developed *event cameras* have the potential to eliminate the aforementioned issues by incorporating novel bio-inspired vision sensors which contain pixels that are asynchronous and work independently from each other [1]. Each pixel is sensitive to local relative light intensity variations, and when this variation exceeds a threshold, they generate signals called *events*, in continuous time. Therefore, the data output from these cameras is a stream of asynchronous events, where each event encodes the pixel location (x, y) and polarity $p \in \{+1, -1\}$ of the intensity change, together with a precise timestamp t. The event stream has a highly varying rate depending on the scene details such as brightness change, motion, and texture. These working principles of event cameras bring many advantages compared to traditional frame-based cameras, such as high dynamic range, high temporal resolution, and low latency. Due to the numerous advantages it offers, event data has been increasingly incorporated into various recognition tasks, including object detection [2], semantic segmentation [3], and fall detection [4]. Furthermore, event data has been utilized in challenging robotic applications that require high-speed perception, such as an object-catching quadrupedal robot [5] and an ornithopter robot capable of avoiding dynamic obstacles [6].

Despite its desirable properties, humans can not directly interpret event streams as we do for intensity images, and

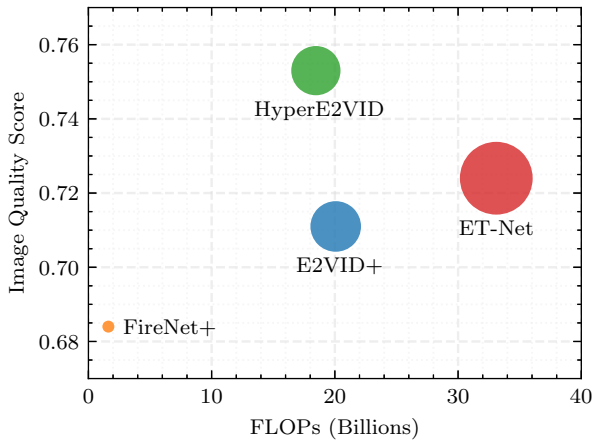


Fig. 1. **Comparison of our HyperE2VID method with state-of-the-art event-based video reconstruction methods based on image quality and computational complexity.** Image quality scores are calculated by normalizing and averaging each of the quantitative scores reported in Table I, where normalization maps the best and worst possible score for each metric to 1.0 and 0.0. Number of floating point operations (FLOPs) are measured as described in Section IV-F. Circle sizes indicate the number of model parameters, as detailed in Table II. The methods with lower image quality scores are not included for clarity of presentation.

BE is with the Department of Computer Engineering, Hacettepe University, TR-06800 Ankara, Turkey, also with HAVELSAN Inc., TR-06510, Ankara, Turkey

OE is with the Department of Computer Engineering, Hacettepe University, TR-06800 Ankara, Turkey, also with HAVELSAN Inc., TR-06510, Ankara, Turkey

CS is with the Department of Computer Engineering, Hacettepe University, TR-06800 Ankara, Turkey, also with ROKETSAN Inc., TR-06780, Ankara, Turkey

AE is with the Department of Computer Engineering, Koc University, TR-34450 Istanbul, Turkey, also with the Koc University Is Bank AI Center, TR-34450, Istanbul, Turkey.

EE is with the Department of Computer Engineering, Hacettepe University, TR-06800 Ankara, Turkey.

This paper has supplementary downloadable material as a single PDF file, available at http://ieeexplore.ieee.org., provided by the author. Contact burakercan@hacettepe.edu.tr for further questions about this work.

high-quality intensity images are the most natural way to understand visual data. Hence, the task of reconstructing intensity images from events has long been a cornerstone in event-based vision literature. Another benefit of reconstructing high-quality intensity images is that one can immediately apply successful frame-based computer vision methods to the reconstruction results to solve various tasks.

Recently, deep learning based methods have obtained impressive results in the task of video reconstruction from events (e.g. [7]–[9]). To use successful deep architectures in conjunction with event-based data, these methods typically group events in time windows and accumulate them into grid-structured representations like 3D voxel grids through which the continuous stream of events is transformed into a series of voxel grid representations. These grid-based representations can then be processed with recurrent neural networks (RNNs), where each of these voxel grids is consumed at each time step.

Since events are generated asynchronously only when the intensity of a pixel changes, the resulting event voxel grid is a sparse tensor, incorporating information only from the changing parts of the scene. The sparsity of these voxel grids is also highly varying. This makes it hard for neural networks to adapt to new data and leads to unsatisfactory video reconstructions that contain blur, low contrast, or smearing artifacts ([7], [8], [10]). Recently, Weng *et al*. [9] proposed to incorporate a Transformer [11] based module to an event-based video reconstruction network in order to better exploit the global context of event tensors. This complex architecture improves the quality of reconstructions, but at the expense of higher inference times and larger memory consumption.

The methods mentioned above try to process the highly varying event data with *static* networks, in which the network parameters are kept fixed after training. Concurrently, there has been a line of research that investigates *dynamic* network architectures that allow the network to adapt its parameters dynamically according to the input supplied at inference time. A well-known example of this approach is the notion of *hypernetworks* [12], which are smaller networks that are used to dynamically generate weights of a larger network at inference time, conditioned on the input. This dynamic structure allows the neural networks to increase their representation power with only a minor increase in computational cost [13].

In this work, we present HyperE2VID which improves the current state-of-the-art in terms of image quality and efficiency (see Fig. 1) by employing a dynamic neural network architecture via hypernetworks. Our proposed model utilizes a main network with a convolutional recurrent encoder-decoder architecture, similar to E2VID [7]. We enhance this network by employing dynamic convolutions, whose parameters are generated dynamically at inference time. These dynamically generated parameters are also spatially varying such that there exists a separate convolutional kernel for each pixel, allowing them to adapt to different spatial locations as well as each input. This spatial adaptation enables the network to learn and use different filters for static and dynamic parts of the scene where events are generated at low and high rates, respectively. We design our hypernetwork architecture in order to avoid the high computational cost of generating per-pixel adaptive filters via filter decomposition as in [14].

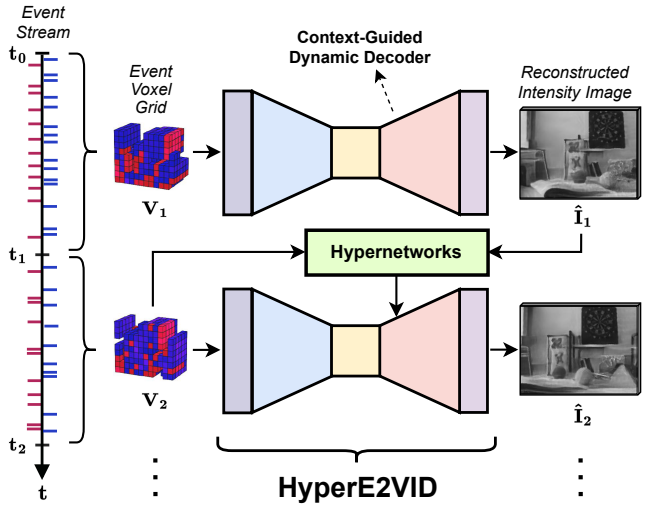


Fig. 2. **HyperE2VID** uses a recurrent encoder-decoder backbone, consuming an event voxel grid at each time step. It enhances this architecture by employing per-pixel, spatially-varying dynamic convolutions at the decoder, whose parameters are generated dynamically at inference time via hypernetworks.

Fig. 2 presents an overview of our proposed method, HyperE2VID, for reconstructing video from events. Our approach is designed to guide the dynamic filter generation through a *context* that represents the current scene being observed. To achieve this, we leverage two complementary sources of information: events and images. We incorporate a context fusion module in our hypernetwork architecture to combine information from event voxel grids and previously reconstructed intensity images. These two modalities complement each other since intensity images capture static parts of the scene better, while events excel at dynamic parts. By fusing them, we obtain a context tensor that better represents both static and dynamic parts of the scene. This tensor is then used to guide the dynamic per-pixel filter generation. We also employ a curriculum learning strategy to train the network more robustly, particularly in the early epochs of training when the reconstructed intensity images are far from optimal.

To the best of our knowledge, this is the first work that explores the use of hypernetworks and dynamic convolutions for event-based video reconstruction. The closest to our work is SPADE-E2VID [10] where the authors employ adaptive feature denormalization in decoder blocks of the E2VID architecture. Rather than feature denormalization, we directly generate per-pixel dynamic filters via hypernetworks for the first decoder block. Specifically, our contributions can be summarized as follows:

- We propose the first dynamic network architecture for the task of video reconstruction from events[1], where we extend existing static architectures with hypernetworks, dynamic convolutional layers, and a context fusion block.
- We show via experiments that this dynamic architecture can generate higher-quality videos than previous state-of-the-art, while also reducing memory consumption and inference time.

[1]Code is available at https://ercanburak.github.io/HyperE2VID.html

II. Related Work

A. Event-Based Video Reconstruction

Reconstructing intensity images from events is a popular topic in event-based vision literature, characterized by a variety of approaches with distinct assumptions and methodologies. Initial efforts in this field often depended on restrictive assumptions such as predetermined or limited camera movement, static scenes, or brightness constancy. More recent advancements, however, leverage deep learning techniques, which naturally integrate image priors into their models during the learning process.

Pioneering works, initiated by Cook et al. [15], typically aimed at simultaneously estimating multiple quantities like intensity images, spatial gradients, and optical flow [16]–[18]. This multi-faceted approach benefits from the dynamic interaction between these elements, as exemplified by the event generation model of Gallego et al. [19], which correlates optical flow, scene gradients, and event data. In methods that primarily predict scene gradients, a common subsequent step involves employing Poisson integration [20] to derive intensity images from these gradients.

Kim *et al*. [16] introduced a filter-based method for estimating scene gradients and ego-motion, but it was limited to rotational camera movements. They later expanded this work in [17] to accommodate free camera motion, though still confined to static scenes. Bardow *et al*. [18] approached dynamic scenes by variational optimization, estimating intensity images and optical flow under the brightness constancy assumption.

Barua *et al*. [21] were the first to show that motion estimation is not necessary for intensity image reconstruction, employing a patch-based dictionary learning method. Following a similar vein, Munda *et al*. [22] proposed an optimization-based method, minimizing an energy function with a data fidelity term based on direct event integration and a manifold regularization term. Scheerlinck *et al*. [23] also used event integration but added a per-pixel temporal high-pass filter to mitigate noise. While their approach allowed for continuous time processing of events, it resulted in artifacts due to the loss of low-frequency information from static backgrounds.

The last few years have witnessed many works that utilize neural networks and deep learning methodologies for the task of intensity image reconstruction. Wang *et al*. [24] represented groups of events with spatio-temporal voxel grids and fed them to a conditional GAN to output intensity images. In their seminal work, Rebecq *et al*. [7] proposed a recurrent fully convolutional network called E2VID to which they input voxel grids of events to produce an intensity image. They trained this network on a large synthetic dataset generated with ESIM [25] using the perceptual loss of [26] and showed that this generalizes well to real event data at test time. As a follow-up study [27], the authors employed temporal consistency loss [28] to minimize temporal artifacts.

After E2VID, many works attempted to enhance it from various perspectives. Scheerlinck *et al*. [29] replaced E2VID architecture with a lightweight recurrent network called FireNet, which has much less memory consumption and faster inference. However, the reconstructions of FireNet were not as good, particularly in scenarios with fast motion. Stoffregen *et al*. [8] improved the results of E2VID and FireNet by matching statistics of synthetic training data to that of real-world test data, resulting in E2VID+ and FireNet+. Cadena *et al*. [10] employed spatially-adaptive denormalization (SPADE) [30] layers in E2VID architecture, improving the quality of reconstructed videos, especially for early frames, but with an increased computational cost. Similarly, Weng *et al*. [9] incorporated a Transformer [11] based module to the CNN-based encoder-decoder architecture of E2VID, improving the reconstruction quality at the expense of increased computational complexity.

In contrast to these, a few recent works followed somewhat different approaches, mainly targeting aspects other than the quality of reconstructions. As an example, Paredes-Vallés and de Croon [31] turned back to the idea of simultaneously estimating optical flow and intensity images via photometric constancy assumption, and suggested a method based on self-supervised learning, eliminating the need for synthetic training data with ground truth frames. Zhu *et al*. [32] used a deep spiking neural network (SNN) architecture, targeting computationally efficient neuromorphic hardware. Zhang *et al*. [33] formulated the event-based image reconstruction task as a linear inverse problem based on optical flow, and suggested a method without training deep neural networks. Although these methods brought improvements in aspects like required training data, computational efficiency, or explainability, the visual quality of their reconstructions was not as strong.

There are also works that target a slightly different task. As an example, Zhang *et al*. [34] argued that the reconstruction performance of E2VID deteriorates when operated with low-light event data, and proposed a novel unsupervised domain adaptation network to generate intensity images as if captured in daylight, from event data of low-light scenes. Mostafavi *et al*. [35] presented a network to generate super-resolved intensity images from events. Similarly, Wang *et al*. [36] introduced a network that can also perform image restoration and super-resolution.

B. Dynamic Networks

Dynamic network is a generic term used to define a network that can adapt its parameters or computational graph dynamically according to its inputs at inference time [13]. This dynamic adaptation can be accomplished in many different ways. For example, one can use a hypernetwork [12], which is a smaller network that is used to dynamically generate weights of a larger network conditioned on the input. For convolutional networks, dynamic filter generation can be position specific as well, such that a different filter is generated for each spatial location and the filtering operation is not translation invariant anymore [37]. Position-specific dynamic filters can be pixel-wise, with a separate kernel for each spatial position, or patch-wise to reduce computational requirements. For example, Nirkin *et al*. proposed HyperSeg, a semantic segmentation network [38] where the encoder generates parameters for dynamic patch-wise convolutional layers in the decoder. In [39], Shaham *et al*. proposed a Spatially-Adaptive Pixel-wise Network (ASAP-Net), where a lightweight convolutional

network acts as a hypernetwork. This hypernetwork works on a lower-resolution input and produces parameters of spatially varying pixel-wise MLPs that process each pixel of the higher-resolution input independently.

It is also possible to dynamically adjust network parameters rather than directly generating them, for example by applying soft attention over multiple convolutional kernels. Both Yang *et al*. [40] and Chen *et al*. [41] proposed to calculate a sample-specific convolutional kernel as a linear combination of many convolutional kernels, where combination coefficients are generated dynamically for each sample. Su *et al*. [42] introduced Pixel-Adaptive Convolution (PAC), where they modify the spatially invariant convolutional kernel by multiplying it with a spatially varying adapting kernel that depends on the input. Chen *et al*. [43] proposed to spatially divide the input feature into regions and process each region with a separate filter. Wang *et al*. [14] proposed Adaptive Convolutions with Dynamic Atoms (ACDA), where they generate sample-specific convolutional filters by multiplying pixel-wise dynamic filter atoms with learned static coefficients. They also decomposed the dynamic atoms to reduce the computational requirements of calculating pixel-wise dynamic filters.

Another approach to dynamic filters is to adapt the shape of the convolutional kernel rather than its parameters. Deformable convolution [44] deforms the geometric structure of the convolutional filter to allow sampling from irregular points. This is achieved by augmenting each sampling location in the filter with dynamic offsets generated by another learned convolutional kernel.

C. Dynamic Networks for Event-Based Vision

Recently, the concept of dynamic networks have started to be used in event-based vision literature as well. In [45], [46] and [47], deformable convolution based feature alignment modules are used for event-based image reconstruction, super-resolution, and HDR imaging, respectively. Vitoria *et al*. [48] used modulated deformable convolutions for the task of event-based image deblurring, where event features encode the motion in the scene, in the form of kernel offsets and modulation masks. Xie *et al*. [49] employed dynamically updated graph CNN to extract discriminative spatio-temporal features for event stream classification.

While the aforementioned methods focus on dynamically changing the computational graphs of networks, there are also works that directly generate network parameters in a dynamic manner. For instance, in the task of event-based video super-resolution, Jing et al. [50] employed a network that takes event representations as inputs and generates parameters for dynamic convolutional layers. In contrast, we employ a context fusion mechanism and generate dynamic parameters guided by both event and image information, motivated by the complementary nature of these two domains. Xiao et al. [51] used dynamic convolutional filters similar to our method but for event-based video frame interpolation. However, they applied each convolutional kernel of shape $1 \times k \times k$ to a specific feature channel to reduce computational demand, which prevents effective modeling of inter-channel dependencies. On the other hand, we consider usual 2D convolutions to let the network model these dependencies, while avoiding high computational costs by using two filter decomposition steps. Furthermore, we utilize previously reconstructed intensity images for context fusion and employ a curriculum learning strategy for robust training, as will be detailed later.

III. The Approach

A. Formulation

Let us assume that we have an event stream $\{e_i\}$ consisting of N_E events that span a duration of T seconds. Each event $e_i = (x_i, y_i, t_i, p_i)$ encodes the location x_i and y_i, the timestamp t_i and the polarity p_i of the ith brightness change that is perceived by the sensor, such that $t_i \in [0, T]$, $p_i \in \{+1, -1\}$, $x_i \in \{0, \dots, W-1\}$ and $y_i \in \{0, \dots, H-1\}$ for all $i \in \{0, \dots, N_E - 1\}$, where W and H are the width and the height of the sensor array, respectively.

Given only these events, our task is to generate an image stream $\{\hat{I}_k\}$ of N_I images from that same time period of T seconds. Each image $\hat{I}_k \in [0,1]^{W \times H}$ is a 2D grayscale representation of the absolute brightness of the scene as if captured by a standard frame-based camera at some time $s_k \in [0, T]$ for all $k \in \{1, \dots, N_I\}$. It is important to note that we constrain our method such that each generated image only depends on past events, *i.e*. only $\{e_i \mid t_i \leq s_k\}$ is used to generate an image $\hat{I}_k$. This allows our method to be used in scenarios where future events are not observed yet, such as reconstructing intensity images from a continuous event camera stream in real-time.

B. Event Representation

Since each event conveys very little information regarding the scene, a common approach in event-based vision literature is to accumulate some number of events into a group, for example by considering a spatio-temporal neighborhood, and then process this group together. We also follow this approach. Assuming that the ground truth intensity frames are available together with the incoming event stream, one can group events such that every event between consecutive frames ends up in the same group. Therefore, given the frame timestamps s_k for all $k \in \{1, \dots, N_I\}$, and letting $s_0 = 0$, the set of events in the kth event group can be defined as follows:

$$G_k \doteq \{e_i \mid s_{k-1} \leq t_i < s_k\} \tag{1}$$

To utilize deep CNN architectures for event-based data, a common choice is to accumulate grouped events into a grid-structured representation such as a voxel grid [52]. Let G_k denote a group of events that spans a duration of ΔT seconds, T_k represent the starting timestamp of that duration, and B be the number of temporal bins that will be used to discretize the timestamps of continuous-time events in the group. The voxel grid $V_k \in \mathbb{R}^{W \times H \times B}$ for that group is formed such that the timestamps of the events from the group are first normalized to the range $[0, B-1]$, and then each event contributes its

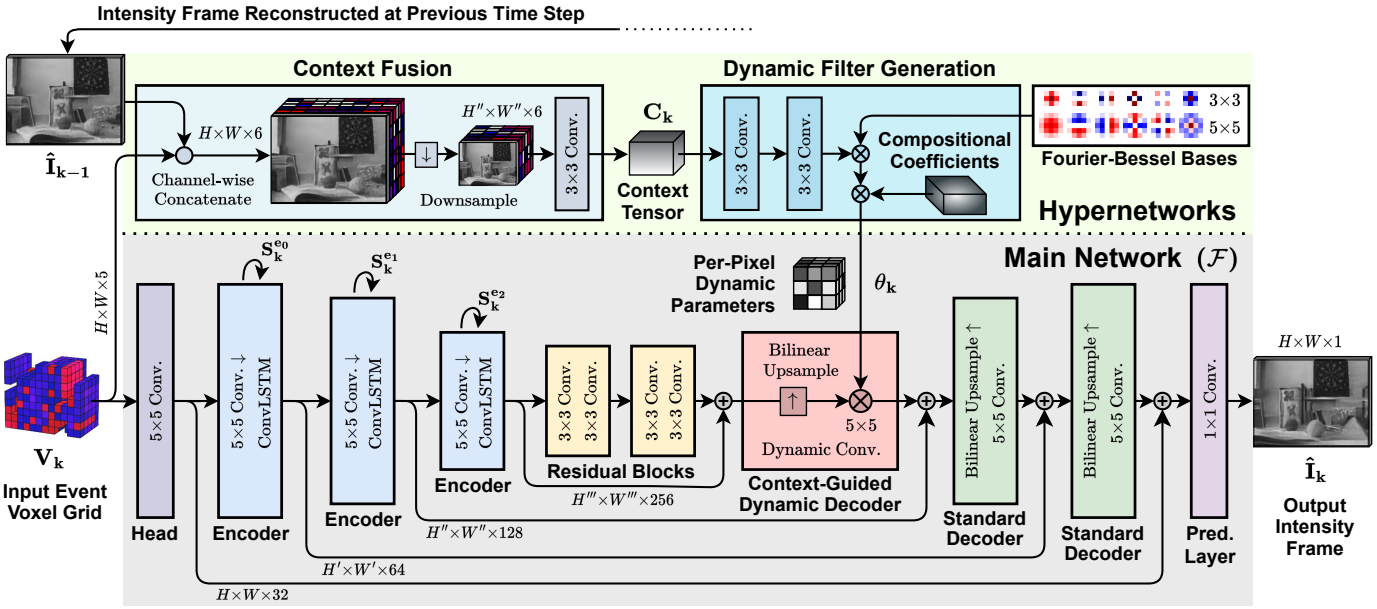


Fig. 3. **Overview of our proposed HyperE2VID architecture.** The main network $\mathcal{F}$ uses a U-Net like architecture to process an event voxel grid V_k and predict the intensity image $\hat{I}_k$ at each time step k. It includes downsampling encoder blocks, upsampling decoder blocks, and skip connections. The encoders incorporate ConvLSTM blocks to capture long temporal dependencies in the sparse event stream. The parameters of the context-guided dynamic decoder (CGDD) block are generated dynamically at inference time, enabling the network to adapt to highly varying event data. These parameters are generated via hypernetworks, consisting of a context fusion (CF) block and a dynamic filter generation (DFG) block. The DFG block employs two filter decomposition steps using multi-scale Fourier-Bessel Bases and learned compositional coefficients, avoiding the high computational cost of per-pixel adaptive filters. The CF block fuses event features from the current time step k with reconstructed image features from the previous time step $k-1$ to generate a context tensor. This fusion scheme combines the dynamic and static parts of the scene captured by events and images, respectively, to generate a context tensor that better represents the overall scene.

polarity to the two temporally closest voxels using a linearly weighted accumulation similar to bilinear interpolation:

$$V_k(x,y,t) = \sum_i p_i \max(0, 1-|t-t_i^*|)\delta(x-x_i, y-y_i) \tag{2}$$

where δ is the Kronecker delta that selects the pixel location, and t_i^* is the normalized timestamp which is calculated as:

$$t_i^* = (B-1)(t_i - T_k)/(\Delta T) \tag{3}$$

where, in all our experiments, we use $B = 5$.

C. HyperE2VID

After representing each event group with a voxel grid, our task is to generate an image stream from the sequence of voxel grids. We use a recurrent neural network that consumes a voxel grid V_k at each time step $k \in \{1, \ldots, N_I\}$, and generates an image $\hat{I}_k$ corresponding to that specific moment. Specifically, we use a U-Net [53] based fully convolutional architecture with recurrent encoder blocks, decoder blocks, and skip connections between them, similar to the E2VID model [7] and the subsequent works of [27], [8], and [10]. Then, we augment this main architecture with hypernetworks, dynamic convolutions, and a context fusion module. We refer to the resulting architecture as HyperE2VID.

Fig. 3 shows an overview of the proposed HyperE2VID framework. Our model consists of a main network $\mathcal{F}$ and hypernetworks that generate parameters for the dynamic part of the main network. From its input to output, the network $\mathcal{F}$ consists of one head layer, three recurrent encoder blocks, two residual blocks, one context-guided dynamic decoder (CGDD) block, two standard decoder blocks, and a prediction layer. The dynamic filter generation (DFG) block and the context fusion (CF) block act as hypernetworks that generate pixel-wise dynamic filter parameters for the dynamic part of the main network, *i.e.* the CGDD block.

More formally, let S_k be the recurrent state of the network for a time step k, containing states $\mathbf{S_k^{e_n}}$ of the three encoder blocks, where $n \in \{0, 1, 2\}$. Given the states from the previous time step, S_{k-1}, and the event voxel grid from the current time step, V_k, the main network $\mathcal{F}$ calculates the current states S_k and predicts the intensity image $\hat{I}_k$ as follows:

$$(\hat{I}_k, S_k) = \mathcal{F}(V_k, S_{k-1}, \theta_k) \tag{4}$$

with θ_k denoting the parameters of the convolutional layer at the CGDD block, which are generated dynamically at inference time by the DFG block, as below:

$$C_k = \text{CF}(V_k, \hat{I}_{k-1}) \tag{5}$$

$$\theta_k = \text{DFG}(C_k) \tag{6}$$

To generate the parameters of the dynamic decoder, we use both the current event voxel grid V_k and the previous reconstruction result $\hat{I}_{k-1}$. The CF block fuses these inputs to generate a context tensor C_k, which is then used by the DFG block. This approach is motivated by the complementary nature of the two domains. Events are better suited for capturing fast motion due to their high temporal resolution but cannot capture static parts of the scene. In contrast, intensity images

are better at capturing static parts of the scene. By fusing V_k and $\hat{I}_{k-1}$, the context tensor C_k incorporates useful features that better describe the static and dynamic parts of the scene.

Skip connections carry output feature maps of the head layer and each encoder block to the inputs of the respective symmetric decoder components, *i.e*. before each decoder block and the prediction layer. Element-wise summation is performed for these skip connections. ReLU activations are used for each convolutional layer unless specified otherwise. We describe each component of our architecture in more detail below:

Head layer. The head layer consists of a convolutional layer with a kernel size of 5. The convolutional layer processes the event voxel grid with 5 temporal channels and outputs a tensor with 32 channels, while the input's spatial dimensions H and W are maintained.

Encoder blocks. Each encoder block consists of a convolutional layer followed by a ConvLSTM [54]. The convolutional layer has a kernel size of 5 and stride of 2, thus, it reduces the spatial dimensions of the input feature map by half. On the other hand, it doubles the number of channels. The ConvLSTM has a kernel size of 3 and maintains the spatial and channel dimensions of its inputs and internal states.

Residual blocks. Each residual block in our network comprises two convolutional layers with a kernel size of 3 that preserve the input's spatial and channel dimensions. A skip connection adds the input features to the output features of the second convolution before the activation function.

Context-Guided Dynamic Decoder (CGDD) block. The CGDD block includes bilinear upsampling to increase the spatial dimensions, followed by a dynamic convolutional layer. The convolution contains 5×5 kernels and reduces the channel size by half. The parameters θ_k of this convolution are generated dynamically during inference time by the DFG block.

It is important to emphasize that all dynamic parameters are generated pixel-wise in that there exists a separate convolutional kernel for each pixel. This spatial adaptation is motivated by the fact that the pixels of an event camera work independently from each other. When there is more motion in one part of the scene, events are generated at a higher rate at corresponding pixels, and the resulting voxel grid is denser in those regions. Our design enables the network to learn and use different filters for each part of the scene according to different motion patterns and event rates, making it more effective to process the event voxel grid with spatially varying densities.

Standard Decoder blocks. Each standard decoder block consists of bilinear upsampling followed by a standard convolutional layer. The details are the same as the context-guided dynamic decoder, except that the parameters are learned at training time and fixed at inference time.

Prediction layer. The prediction layer is a standard convolutional layer with a kernel size of 1, and it outputs the final predicted intensity image with 1 channel. We do not use an activation function after this layer.

Dynamic Filter Generation (DFG) block. A crucial component of our method is the dynamic filter generation. This block consumes a *context* tensor and output parameters for the CGDD block. The context tensor C_k is expected to be at the same spatial size as the input of the dynamic convolution ($W'' \times H''$). To generate the context tensor, we use a context fusion mechanism that fuses features from the event voxel grid (V_k) and the previous reconstruction ($\hat{I}_{k-1}$) of the network.

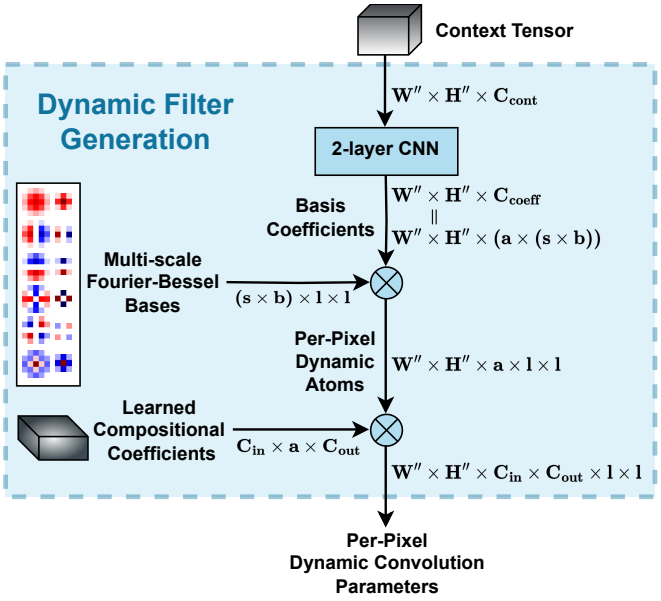


Fig. 4. **Dynamic Filter Generation (DFG) block**. DFG block takes a context tensor as input and generates per-pixel dynamic convolution parameters via two filter decomposition steps, making use of pre-fixed multi-scale Fourier-Bessel bases and learned compositional coefficients. More details are given in Section III-C.

To reduce the computational cost, we use two filter decomposition steps while generating per-pixel dynamic filters. First, we decompose filters into per-pixel filter atoms generated dynamically. Second, we further decompose each filter atom as a truncated expansion with pre-fixed multi-scale Fourier-Bessel bases. Inspired by ACDA [14], our approach generates efficient per-pixel dynamic convolutions that vary spatially. However, unlike ACDA, our network architecture performs dynamic parameter generation independently through hypernetworks, which are guided by a context tensor designed to provide task-specific features for event-based video reconstruction.

Fig. 4 illustrates the detailed operations of our proposed DFG block. A context tensor with dimensions $W'' \times H'' \times C_{cont}$ is fed into a 2-layer CNN, producing pixel-wise basis coefficients of size C_{coeff} that are used to generate per-pixel dynamic atoms via pre-fixed multi-scale Fourier-Bessel bases. These bases are represented by a tensor of size $s \times b \times l \times l$, where s is the number of scales, b is the number of Fourier-Bessel bases at each scale, and l is the kernel size for which the dynamic parameters are being generated. Multiplying the multi-scale Fourier-Bessel bases with the basis coefficients generate per-pixel dynamic atoms of size $l \times l$. Number of generated atoms for each pixel is a, so it is possible to represent all of the generated atoms by a tensor of size $W'' \times H'' \times a \times l \times l$. Next, the compositional coefficients tensor of size $C_{in} \times a \times C_{out}$ is multiplied with these per-pixel dynamic atoms. These learned coefficients are fixed at inference time and shared across spatial positions. This multiplication produces a tensor of size

$W' \times H' \times C_{in} \times C_{out} \times l \times l$, which serves as the parameters for the per-pixel dynamic convolution. Here, C_{in} and C_{out} are the number of input and output channels for the dynamic convolution, respectively. For the DFG block, we set $a = 6$, $b = 6$, and $l = 5$. The number of scales $s = 2$, meaning that we use 3×3 and 5×5 sized Fourier-Bessel bases. Since we have $b = 6$ bases at each scale, we have a total of $s \times b = 12$ Fourier-Bessel bases. The 2-layer CNN has a hidden channel size of 64. Both convolutional layers have a kernel size of 3, and they are followed by a batch normalization [55] layer and a tanh activation. The output of the CNN has $C_{coeff} = a \times b \times s = 72$ channels, to produce a separate coefficient per dynamic atom and per Fourier-Bessel basis.

Context Fusion (CF) block. The events are generated asynchronously only when the intensity of a pixel changes, and therefore the resulting event voxel grid is a sparse tensor, incorporating information only from the changing parts of the scene. Our HyperE2VID architecture conditions the dynamic decoder block parameters with both the current event voxel grid V_k and the previous network reconstruction $\hat{I}_{k-1}$. These two domains provide complementary information; the intensity image is better suited for static parts of the scene, while the events are better for dynamic parts. We use the CF block to fuse this information, enabling the network to focus on intensity images for static parts and events for dynamic parts. Our context fusion block design concatenates V_k and $\hat{I}_{k-1}$ channel-wise to form a 6-channel tensor. We downsample this tensor to match the input dimensions of the dynamic convolution at the CGDD block and then use a 3×3 convolution to produce a context tensor with 32 channels. While more complex architectures are possible, we opt for a simple design for the context fusion block.

D. Training Details

During training, we employ the following loss functions:

Perceptual Reconstruction Loss. We use the AlexNet [56] variant of the learned perceptual image patch similarity (LPIPS) [26] to enforce reconstructed images to be perceptually close to ground truth intensity images. LPIPS works by passing the predicted and reference images through a deep neural network architecture that was trained for visual recognition tasks, and using the distance between deep features from multiple layers of that network as a measure of the perceptual difference between the two images.

$$\mathcal{L}_k^{\text{LPIPS}} = \text{LPIPS}(\hat{I}_k, I_k) \tag{7}$$

Temporal Consistency Loss. We use the *short-term temporal loss* of [28], as employed in [27], to enforce temporal consistency between the images that are reconstructed in consecutive time steps of the network. This loss works by warping the previously reconstructed image using a ground truth optical flow to align it with the current reconstruction and using a masked distance between these aligned images as a measure of temporal consistency, where the mask is calculated from the warping error between the previous and the current ground truth intensity images. More formally, the temporal consistency loss is calculated as:

$$\mathcal{L}_k^{\text{TC}} = M_k \|\hat{I}_k - W(\hat{I}_{k-1}, F_{k \to k-1})\|_1 \tag{8}$$

where $F_{k \to k-1}$ denotes the optical flow map between time steps k and $k-1$, W is the warping function, and M_k represents the occlusion mask which is computed as:

$$M_k = \exp(-\alpha \|I_k - W(I_{k-1}, F_{k \to k-1})\|_2^2) \tag{9}$$

where we use $\alpha = 50$ as in [27], [28]. The mask M_k contains smaller terms for pixels where the warping error between consecutive ground truth images is high, and therefore the masking operation effectively discards these pixels from the temporal consistency calculation of reconstructed frames.

The final loss for a time step k is the sum of the perceptual reconstruction and temporal losses:

$$\mathcal{L}_k = \mathcal{L}_k^{\text{LPIPS}} + \mathcal{L}_k^{\text{TC}} \tag{10}$$

During training, we calculate the loss $\mathcal{L}_k$ at every T_S time-steps in a training sequence, and the gradients of this loss with respect to the network parameters are calculated using the Truncated Back-propagation Through Time (TBPTT) algorithm [57] with a truncation period of T_T time-steps. Setting $T_S > 1$ and $T_T < k$ reduces memory requirements and speeds up the training process.

We implement our network in PyTorch [58]. We train the recurrent network with sequences of length 40, with the network parameters initialized using He initialization [59]. At the first time step of each sequence, the initial values of the previous reconstruction, $\hat{I}_0$, and the network states, S_0, are set to zero tensors. The loss calculation and the truncation periods are set as $T_S = 10$ and $T_T = 5$, respectively. We train our network for 400 epochs using a batch size of 10 and the AMSGrad [60] variant of the Adam [61] optimizer with a learning rate of 0.001. To track our trainings and experimental analyses, we use Weights & Biases [62].

At the start of the training, the previous reconstruction $\hat{I}_{k-1}$ of the network, which is used for context fusion, is far from optimal. This makes it harder for the context fusion block to learn useful representations, especially in the earlier epochs of the training. To resolve this issue, we employ a curriculum learning [63] strategy during the training. We start the training by using the ground truth previous image I_{k-1} instead of the previous reconstruction of the network $\hat{I}_{k-1}$ for context fusion. For the first 100 epochs, we gradually switch to using images that the network reconstructs at the previous time step, by weighted averaging them with ground-truth images. After the 100th epoch, we continue the training by only using previous reconstructions for context fusion. Therefore, we use a modified version of Equation (5) during training:

$$\beta = \min(1, \frac{\textit{epoch}}{100}) \tag{11}$$

$$I_{\textit{context}} = \beta \cdot \hat{I}_{k-1} + (1 - \beta) \cdot I_{k-1} \tag{12}$$

$$C_k = \text{CF}(V_k, I_{\textit{context}}) \tag{13}$$

This curriculum learning strategy allows the parameters of the hypernetworks to be learned more robustly, enabling the training process to converge to a better-performing model.

During training, we augment the images and event tensors with random crops and flips as suggested in [27]. The size of random crops is 112×112, and the probability of vertical and horizontal flips are both 0.5. Furthermore, we employ dynamic train-time noise augmentation, pause augmentation, and hot-pixel augmentation as described in [8].

IV. Experimental Analysis

A. Training Dataset

We generate a synthetic training set as described in [8], using the `Multi-Objects-2D` renderer option of ESIM [25] where multiple moving objects are captured with a camera restricted to 2D motion. The dataset consists of 280 sequences, all of which are 10 secs in length. The contrast threshold values for event generation are in the range of 0.1 to 1.5. Each sequence includes generated event streams together with ground truth intensity images and optical flow maps with an average rate of $51\,\mathrm{Hz}$. The resolutions of event and frame cameras are both 256×256. The sequences include scenes containing up to 30 foreground objects with varying speeds and trajectories, where the objects are randomly selected images from the MS-COCO dataset [64].

B. Testing Datasets

To comprehensively evaluate our method, we utilize sequences from five real-world datasets, each selected for their unique characteristics and relevance to different aspects of event-based video reconstruction. These datasets are the Event Camera Dataset (ECD) [65], the Multi Vehicle Stereo Event Camera (MVSEC) dataset [66], the High-Quality Frames (HQF) dataset [8], the UZH-FPV Drone Racing (FPVDR) dataset [67], and the Color Event Camera Dataset (CED) [68].

The ECD dataset, with its DAVIS240C sensor-generated frames and events, is pivotal for evaluating reconstructions in environments with 6-DOF camera movement and varying speeds. Within this dataset, we introduce the FAST subset to specifically assess reconstruction quality under conditions of rapid camera motion. The MVSEC dataset offers longer sequences in both indoor and outdoor settings, captured by DAVIS 346B cameras. This dataset is integral for analyzing performance in diverse environments. Additionally, we derive the NIGHT subset from MVSEC to evaluate our method's effectiveness in low-light conditions, a challenging scenario for event-based reconstruction. The HQF dataset provides a variety of indoor and outdoor sequences with well-exposed and minimally blurred frames, crucial for benchmarking reconstruction quality in more controlled environments. The UZH-FPV Drone Racing dataset, with its fast and aggressive drone movements, is ideal for testing our method under extreme motion conditions, offering a rigorous assessment of reconstruction capabilities in dynamic scenarios. Lastly, the CED dataset's color frames and events, captured with the Color-DAVIS346 camera, allow us to demonstrate our method's color reconstruction ability, particularly in scenes with vibrant colors and challenging lighting conditions.

Detailed descriptions of these datasets and their specific usage in our analysis are given in the supplementary material.

C. Evaluation Metrics

We evaluate the methods using three full-reference evaluation metrics, mean squared error (MSE), structural similarity (SSIM) [69], and learned perceptual image patch similarity (LPIPS) [26] when high-quality, distortion-free ground truth frames are available. To assess image quality under challenging scenarios, such as low-light and fast motion, where ground truth frames are of low quality, we use a no-reference metric, BRISQUE [70]. These metrics have some settings that affect their results, and thus we provide the implementation details of them in the supplementary material to facilitate reproducibility.

D. Competing Approaches

We compare our method against seven other methods from the literature, which are E2VID [27], FireNet [29], FireNet+ and E2VID+ [8], SPADE-E2VID [10], SSL-E2VID [31], and ET-Net [9]. E2VID+ and SSL-E2VID use the same network architecture as E2VID, but their training details are different. Similarly, FireNet+ uses the same network architecture as FireNet. We use the pre-trained models that the respective authors publicly share for each of these methods, and evaluate them using the same datasets and under the same settings. All of these methods use the same voxel grid event representation as ours (Section III-B). We group events that have timestamps between every two consecutive ground truth frames and form the voxel grids using these. We also apply any pre-processing and post-processing steps when required by the method, such as the event tensor normalization and robust min/max normalization of E2VID. After generating reconstructions for each method, we perform quantitative analysis using the full-reference metrics, MSE, SSIM, and LPIPS, or the no-reference metric BRISQUE, depending on whether high-quality ground truth frames are available or not. We do not perform histogram equalization to reconstructions or ground truth images before calculating evaluation metrics. The quantitative results and the qualitative analysis are given in Section IV-E, as well as color reconstruction results for sample scenes from the CED dataset. We also compare the computational complexity of each network architecture in Section IV-F.

E. Experimental Results

Table I presents the quantitative results obtained from evaluating the methods on sequences from the aforementioned datasets. We calculate the average values of each metric across all evaluated frames. The HyperE2VID method achieves state-of-the-art performance in terms of most metrics. On the ECD and MVSEC datasets, it outperforms the second-best method, ET-Net, by a large margin. On the HQF dataset, it delivers results on par with state-of-the-art approaches. In challenging scenarios involving fast camera motion (FAST and FPVDR), it obtains the best BRISQUE scores; and in night driving sequences (NIGHT), it obtains the second-best BRISQUE scores after E2VID, surpassing all the other methods. These results demonstrate the effectiveness of the proposed HyperE2VID method, which generates perceptually more pleasing and high-fidelity reconstructions.

TABLE I
QUANTITATIVE RESULTS OF EXISTING METHODS AND OUR PROPOSED METHOD ON SEQUENCES FROM ECD, MVSEC, HQF, AND FPVDR DATASETS.

	ECD			MVSEC			HQF			FAST	NIGHT	FPVDR
	MSE ↓	SSIM ↑	LPIPS ↓	MSE ↓	SSIM ↑	LPIPS ↓	MSE ↓	SSIM ↑	LPIPS ↓	BRISQUE ↓		
E2VID [27]	0.179	0.450	0.322	0.225	0.241	0.644	0.098	0.468	0.371	<u>14.957</u>	**2.153**	<u>14.239</u>
FireNet [29]	0.131	0.459	0.320	0.292	0.199	0.700	0.094	0.423	0.441	19.957	21.311	21.395
E2VID+ [8]	0.070	0.503	0.236	0.132	0.262	0.514	0.036	<u>0.533</u>	**0.252**	22.627	12.285	18.677
FireNet+ [8]	0.063	0.452	0.290	0.218	0.212	0.570	0.040	0.471	0.314	18.399	10.019	15.502
SPADE-E2VID [10]	0.091	0.461	0.337	0.138	0.266	0.589	0.077	0.400	0.502	18.925	24.011	21.248
SSL-E2VID [31]	0.092	0.415	0.380	0.124	0.264	0.693	0.082	0.421	0.467	46.199	49.562	59.454
ET-Net [9]	<u>0.047</u>	<u>0.552</u>	<u>0.224</u>	<u>0.107</u>	<u>0.288</u>	<u>0.489</u>	<u>0.032</u>	**0.534**	0.260	19.698	15.533	22.745
HyperE2VID (ours)	**0.033**	**0.576**	**0.212**	**0.076**	**0.315**	**0.476**	**0.031**	0.530	<u>0.257</u>	**14.024**	<u>5.973</u>	**14.178**

We present qualitative results for ECD, MVSEC, HQF, and FPVDR datasets in Fig. 5. We omit reconstructions of FireNet and SPADE-E2VID due to lower quantitative scores and focus on the performances of E2VID, E2VID+, FireNet+, E2VID+, SSL-E2VID, ET-Net, and HyperE2VID. Sample scenes are shown from ECD (rows 1,2), MVSEC (rows 3,4), HQF (rows 5-7), and FPVDR (row 8) datasets, as well as fast parts of ECD (FAST, row 9) and night sequences of MVSEC (NIGHT, row 10) datasets. Each row shows reconstructions of each model (first six columns) with the reference frame given in the rightmost column.

The visual qualities of reconstructions are mostly in line with the quantitative results. Among the six methods, FireNet+ and SSL-E2VID tend to have the lowest quality, with prominent visual artifacts and blurry regions. Reconstructions of E2VID+ have fewer artifacts, especially at scenes from the HQF dataset. E2VID+ also produces nice-looking images for the outdoor scenes of the MVSEC dataset. However, its reconstructions are generally of low contrast and blurry around the edges. ET-Net has better contrast but has more artifacts at textureless regions and around the edges of objects. The reconstructions of HyperE2VID are of high contrast and sharp around the edges. Moreover, the textureless regions are mostly reconstructed with fewer artifacts.

These qualitative results show that the reconstructions from most methods display artifacts to varying degrees. In light of this widespread issue, we present a post-processing framework in the supplementary material. This framework, which can be applied to reconstructions from any event-based video reconstruction method, aims to eliminate or significantly reduce various types of artifacts, particularly in textureless regions.

In Fig. 6, we show color reconstructions from HyperE2VID alongside those from two top-performing competitors, E2VID+ and ET-Net, using sample scenes from the CED dataset. These are compared with reference frames from the Color-DAVIS346 camera. To generate these color reconstructions, we adopt the method described in [27]. This involve reconstructing each color channel separately at quarter resolution, then upsampling and merging them to form a full-color image. Next, we convert this image to LAB color space and replace its luminance channel with a high-resolution grayscale reconstruction derived from all events. The color results, as shown in Fig. 6, demonstrate HyperE2VID's ability to produce color images of superior quality. These images exhibit sharp edges, minimal artifacts, and authentic colors, even in challenging lighting conditions, such as the high-dynamic-range (HDR) scene displayed in the last row.

In our supplementary material, we present comprehensive ablation studies and additional analyses of the HyperE2VID model. Key design components like context-guided per-pixel dynamic convolutions, hypernetworks, and context fusion are rigorously evaluated to affirm their impact. We specifically explore the adaptability of our method in varied scenarios including slow motion, fast motion, and low-light conditions, highlighting the critical role of contextually relevant information in these environments. Our investigation also includes the impact of varying temporal windows and event counts in constructing event voxel grids, highlighting the versatility of the HyperE2VID architecture across diverse settings. Moreover, we demonstrate its effectiveness in two particularly demanding situations: generating high frame rate videos ranging from $200\,\mathrm{Hz}$ to $5\,\mathrm{kHz}$, and reconstructing scenes during motionless intervals. Our findings not only validate our design choices but also offer valuable directions for future enhancements.

F. Computational Complexity

We also analyze the computational complexity of our method and compare it to other competing methods from the literature. We consider three computational metrics for this analysis: (1) the number of model parameters, (2) the number of floating point operations (FLOPs), and (3) inference time. The number of parameters is an important metric that indicates the memory requirements of the model, while FLOPs specify the computational requirements and efficiency, and finally, the inference time is a direct indicator of the real-time performance of (the maximum frame-per-seconds that can be obtained with) the model. We use data with a resolution of 240×180 to measure FLOPs and inference time, where the average inference times are calculated on a workstation with Quadro RTX 5000 GPU. We present the results of these computational complexity metrics in Table II. Here, the numbers of model parameters are given in millions, FLOPs are given in billions (as GFLOPs), and the inference times are given in milliseconds. Methods that share a common network architecture are presented in the same row. Here, it can be seen that our method provides a good trade-off between accuracy

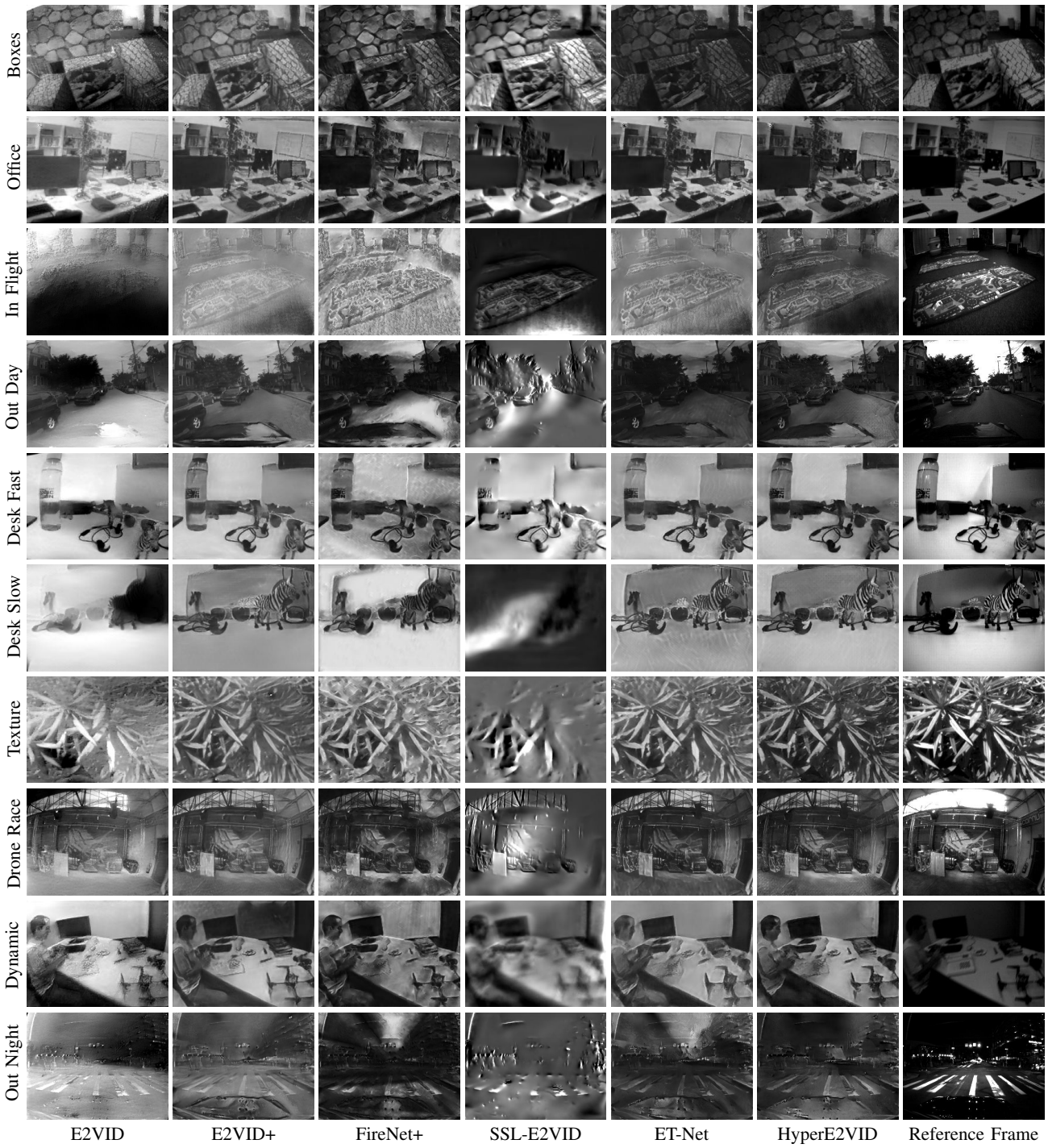


Fig. 5. **Qualitative comparisons on some sequences from ECD (rows 1-2), MVSEC (rows 3-4), HQF (rows 5-7), FPVDR (row 8), FAST (row 9), and NIGHT (row 10).** While the competing approaches suffer from low contrast, blur, and extensive artifacts, HyperE2VID reconstructions have high contrast and preserve sharp details around the edges, with minimal artifacts in textureless regions.

and efficiency. HyperE2VID is a significantly smaller and faster network than ET-Net while generating reconstructions with better visual quality. On the other hand, the smallest and fastest methods, FireNet and FireNet+, generate reconstructions with significantly lower visual quality.

V. Conclusion

In this work, we present HyperE2VID, a novel dynamic network architecture for event-based video reconstruction that improves the state-of-the-art by employing hypernetworks and dynamic convolutions. Our approach generates adaptive filters using hypernetworks, which are dynamically generated at inference time based on the scene context encoded via event voxel grids and previously reconstructed intensity images, and thus deals with static and dynamic parts of the scene more effectively. Experimental results on several challenging datasets show that HyperE2VID outperforms previous state-of-the-art methods in terms of visual quality while reducing

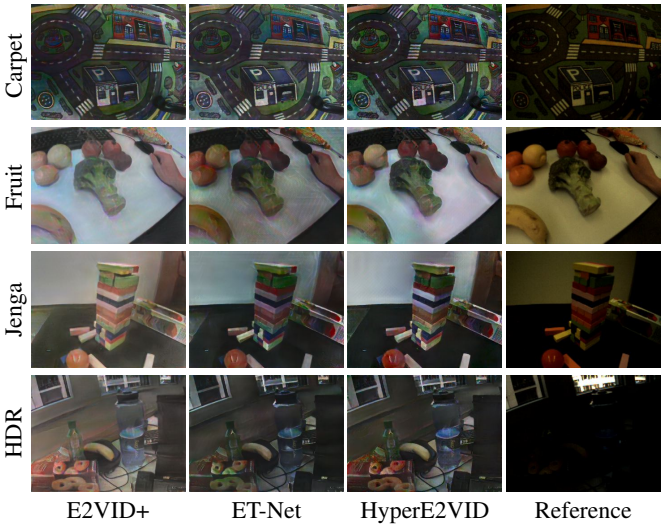


Fig. 6. **Color image reconstructions on CED.** HyperE2VID excels in reconstructing visually appealing scenes from the CED dataset, including those with colorful objects and HDR scenarios, outperforming E2VID+ and ET-Net in visual quality.

TABLE II
COMPUTATIONAL COMPLEXITY OF NETWORK ARCHITECTURES IN TERMS OF THE NUMBER OF MODEL PARAMETERS (IN MILLIONS), NUMBER OF FLOATING POINT OPERATIONS (FLOPS - IN BILLIONS), AND INFERENCE TIME (IN MILLISECONDS).

Network Architecture	Number of Params (M)	GFLOPs	Inference Time (ms)
E2VID [8], [27], [31]	10.71	20.07	<u>5.1</u>
FireNet [8], [29]	**0.04**	**1.62**	**1.6**
SPADE-E2VID [10]	11.46	68.06	16.1
ET-Net [9]	22.18	33.10	32.1
HyperE2VID (ours)	<u>10.15</u>	<u>18.46</u>	6.6

memory consumption, FLOPs, and inference time. Our work demonstrates the potential of dynamic network architectures and hypernetworks for processing highly varying event data, opening up possibilities for future research in this direction, targeting more tasks like event-based optical-flow estimation.

VI. ACKNOWLEDGMENTS

This work was supported in part by KUIS AI Research Award, TUBITAK-1001 Program Award No. 121E454, Hacettepe University BAP Coordination Unit with grant no. FHD-2023-20611, and BAGEP 2021 Award of the Science Academy to A. Erdem. We thank all the reviewers for their valuable comments.

REFERENCES

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2020.

[2] J. Li, J. Li, L. Zhu, X. Xiang, T. Huang, and Y. Tian, "Asynchronous spatio-temporal memory network for continuous event-based object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 2975–2987, 2022.

[3] Z. Jia, K. You, W. He, Y. Tian, Y. Feng, Y. Wang, X. Jia, Y. Lou, J. Zhang, G. Li *et al.*, "Event-based semantic segmentation with posterior attention," *IEEE Trans. Image Process.*, 2023.

[4] G. Chen, S. Qu, Z. Li, H. Zhu, J. Dong, M. Liu, and J. Conradt, "Neuromorphic vision-based fall localization in event streams with temporal–spatial attention weighted network," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9251–9262, 2022.

[5] B. Forrai, T. Miki, D. Gehrig, M. Hutter, and D. Scaramuzza, "Event-based agile object catching with a quadrupedal robot," *arXiv preprint arXiv:2303.17479*, 2023.

[6] J. P. Rodríguez-Gómez, R. Tapia, M. d. M. G. Garcia, J. R. Martínez-de Dios, and A. Ollero, "Free as a bird: Event-based dynamic sense-and-avoid for ornithopter robot flight," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5413–5420, 2022.

[7] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *CVPR*, 2019, pp. 3857–3866.

[8] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, "Reducing the sim-to-real gap for event cameras," in *ECCV*, 2020.

[9] W. Weng, Y. Zhang, and Z. Xiong, "Event-based video reconstruction using transformer," in *ICCV*, 2021, pp. 2563–2572.

[10] P. R. G. Cadena, Y. Qian, C. Wang, and M. Yang, "SPADE-E2VID: Spatially-adaptive denormalization for event-based video reconstruction," *IEEE Trans. Image Process.*, vol. 30, pp. 2488–2500, 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[12] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[13] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[14] Z. Wang, Z. Miao, J. Hu, and Q. Qiu, "Adaptive convolutions with per-pixel dynamic filter atom," in *ICCV*, 2021, pp. 12 302–12 311.

[15] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, "Interacting maps for fast visual interpretation," in *IJCNN*, 2011, pp. 770–776.

[16] H. Kim, A. Handa, R. Benosman, S. Ieng, and A. Davison, "Simultaneous mosaicing and tracking with an event camera," in *BMVC*, 2014.

[17] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *ECCV*, 2016, pp. 349–364.

[18] P. Bardow, A. J. Davison, and S. Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *CVPR*, 2016, pp. 884–892.

[19] G. Gallego, C. Forster, E. Mueggler, and D. Scaramuzza, "Event-based camera pose tracking using a generative event model," *arXiv preprint arXiv:1510.01972*, 2015.

[20] A. Agrawal, R. Chellappa, and R. Raskar, "An algebraic approach to surface reconstruction from gradient fields," in *ICCV*, vol. 1, 2005, pp. 174–181.

[21] S. Barua, Y. Miyatani, and A. Veeraraghavan, "Direct face detection and video reconstruction from event cameras," in *WACV*, 2016, pp. 1–9.

[22] G. Munda, C. Reinbacher, and T. Pock, "Real-time intensity-image reconstruction for event cameras using manifold regularisation," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1381–1393, 2018.

[23] C. Scheerlinck, N. Barnes, and R. Mahony, "Continuous-time intensity estimation using event cameras," in *ACCV*, December 2018, pp. 308–324. [Online]. Available: https://cedric-scheerlinck.github.io/files/2018_scheerlinck_continuous-time_intensity_estimation.pdf

[24] L. Wang, S. M. Mostafavi, Y.-S. Ho, K.-J. Yoon *et al.*, "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," in *CVPR*, 2019, pp. 10 081–10 090.

[25] H. Rebecq, D. Gehrig, and D. Scaramuzza, "ESIM: an open event camera simulator," in *CoRL*, 2018, pp. 969–982.

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[27] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[28] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *ECCV*, 2018, pp. 170–185.

[29] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. Mahony, and D. Scaramuzza, "Fast image reconstruction with an event camera," in *WACV*, 2020, pp. 156–163.

[30] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019, pp. 2337–2346.

[31] F. Paredes-Vallés and G. C. de Croon, "Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy," in *CVPR*, 2021, pp. 3446–3455.

[32] L. Zhu, X. Wang, Y. Chang, J. Li, T. Huang, and Y. Tian, "Event-based video reconstruction via potential-assisted spiking neural network," in *CVPR*, 2022, pp. 3594–3604.

[33] Z. Zhang, A. Yezzi, and G. Gallego, "Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 01, pp. 1–18, 2022.

[34] S. Zhang, Y. Zhang, Z. Jiang, D. Zou, J. Ren, and B. Zhou, "Learning to see in the dark with events," in *ECCV*, vol. 1, 2020, p. 2.

[35] S. M. Mostafavi, J. Choi, and K.-J. Yoon, "Learning to super resolve intensity images from events," in *CVPR*, 2020, pp. 2768–2776.

[36] L. Wang, T.-K. Kim, and K.-J. Yoon, "EventSR: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning," in *CVPR*, 2020, pp. 8315–8325.

[37] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," *NeurIPS*, vol. 29, 2016.

[38] Y. Nirkin, L. Wolf, and T. Hassner, "HyperSeg: Patch-wise hypernetwork for real-time semantic segmentation," in *CVPR*, 2021, pp. 4061–4070.

[39] T. R. Shaham, M. Gharbi, R. Zhang, E. Shechtman, and T. Michaeli, "Spatially-adaptive pixelwise networks for fast image translation," in *CVPR*, 2021, pp. 14 882–14 891.

[40] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," *NeurIPS*, vol. 32, 2019.

[41] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *CVPR*, 2020, pp. 11 030–11 039.

[42] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, "Pixel-adaptive convolutional neural networks," in *CVPR*, 2019, pp. 11 166–11 175.

[43] J. Chen, X. Wang, Z. Guo, X. Zhang, and J. Sun, "Dynamic region-aware convolution," in *CVPR*, 2021, pp. 8064–8073.

[44] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017, pp. 764–773.

[45] Y. Zou, Y. Zheng, T. Takatani, and Y. Fu, "Learning to reconstruct high speed and high dynamic range videos from events," in *CVPR*, 2021, pp. 2024–2033.

[46] J. Han, Y. Yang, C. Zhou, C. Xu, and B. Shi, "Evintsr-net: Event guided multiple latent frames reconstruction and super-resolution," in *ICCV*, 2021, pp. 4882–4891.

[47] N. Messikommer, S. Georgoulis, D. Gehrig, S. Tulyakov, J. Erbach, A. Bochicchio, Y. Li, and D. Scaramuzza, "Multi-bracket high dynamic range imaging with event cameras," in *CVPR*, 2022, pp. 547–557.

[48] P. Vitoria, S. Georgoulis, S. Tulyakov, A. Bochicchio, J. Erbach, and Y. Li, "Event-based image deblurring with dynamic motion awareness," in *ECCVW*, 2023, pp. 95–112.

[49] B. Xie, Y. Deng, Z. Shao, H. Liu, and Y. Li, "Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification," *IEEE Robot. Autom. Lett.*, vol. 7, pp. 1976–1983, 2022.

[50] Y. Jing, Y. Yang, X. Wang, M. Song, and D. Tao, "Turning frequency to resolution: Video super-resolution via event cameras," in *CVPR*, 2021, pp. 7772–7781.

[51] Z. Xiao, W. Weng, Y. Zhang, and Z. Xiong, "Eva^2: Event-assisted video frame interpolation via cross-modal alignment and aggregation," *IEEE Trans. Comput. Imaging*, vol. 8, pp. 1145–1158, 2022.

[52] A. Zihao Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based optical flow using motion compensation," in *ECCV*, 2018.

[53] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[54] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *NeurIPS*, vol. 28, pp. 802–810, 2015.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[57] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.

[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, vol. 32, pp. 8026–8037, 2019.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[60] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[62] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[63] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, pp. 41–48.

[64] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[65] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.

[66] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, 2018.

[67] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *ICRA*, 2019, pp. 6713–6719.

[68] C. Scheerlinck, H. Rebecq, T. Stoffregen, N. Barnes, R. Mahony, and D. Scaramuzza, "CED: color event camera dataset," in *CVPRW*, 2019.

[69] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[70] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012.

Supplementary Material for "HyperE2VID: Improving Event-Based Video Reconstruction via Hypernetworks"

Burak Ercan, Onur Eker, Canberk Saglam, Aykut Erdem, *Senior Member, IEEE*, Erkut Erdem, *Senior Member, IEEE*,

In this supplementary document, we provide additional material to complement the main paper. First, we provide comprehensive information about the datasets employed in our evaluations and their specific roles in our analysis. Following that, we offer the implementation details of the evaluation metrics to aid in reproducibility. Next, we delve into the detailed results of our ablation studies and further analyses. Those involve assessing various design elements of HyperE2VID and analyzing reconstruction performance in diverse scenarios, such as employing different event grouping strategies, generating high frame rate videos, and reconstructing video frames during motionless periods. Finally, we present a post-processing framework that can optionally be applied to reconstructions of any event-based video reconstruction method, eliminating or minimizing various types of artifacts encountered in textureless regions.

I. Testing Datasets

To comprehensively evaluate our method, we utilize sequences from five real-world datasets, namely the Event Camera Dataset (ECD) [1], the Multi Vehicle Stereo Event Camera (MVSEC) dataset [2], the High-Quality Frames (HQF) dataset [3], the UZH-FPV Drone Racing dataset [4], and the Color Event Camera Dataset (CED) [5]. We provide the details of these datasets and describe their significance in our analysis below.

Event Camera Dataset (ECD). This dataset is captured by a DAVIS240C sensor [6] where events and frames are generated from the same pixel array of 240×180 resolution. Following the common practice established by Rebecq *et al.* [7], we use seven short sequences from this dataset, where the camera moves with 6-DOF and with increasing speed in six of them. These sequences mostly contain simple office environments with static objects. The ground truth intensity frames are available at an average rate of $22\,\mathrm{Hz}$, and we exclude scores from the initial few seconds of each sequence to align with prior work. Additionally, we exclude the parts of the sequences that contain motion blur due to fast camera motion while evaluating with full-reference metrics. In total, we use 1853 ground truth frames for full-reference metrics, and the specific start and end times of evaluation intervals are provided in [3]. We report these evaluation scores under the name **ECD** in our quantitative results tables. To assess the quality of the reconstructions under fast camera motion, we conduct a separate evaluation using the latter parts of the ECD sequences and a no-reference metric. It comprises a total of 4453 reconstructed frames, and its purpose is to examine the reconstruction quality when the camera moves rapidly. We report the scores from this evaluation under the name **FAST** in the quantitative results tables.

Multi Vehicle Stereo Event Camera (MVSEC) dataset. This dataset has longer sequences of indoor and outdoor environments captured by a pair of DAVIS 346B cameras. These cameras generate events and frames from the same pixel array, which has a resolution of 346×260. We use the data from the left DAVIS camera in our experiments. Following [3], we also use specific time intervals of 6 sequences. Four of them are indoor sequences that are taken from a flying hexacopter, while the two outdoor sequences are taken from a vehicle driving in daylight. The average rate of ground truth intensity frames is around $30\,\mathrm{Hz}$ for indoor sequences and $45\,\mathrm{Hz}$ for outdoor sequences. The specific start and end times of evaluation intervals are given at [3]. The total number of ground truth frames used for evaluation is 11312. We report these scores under the name **MVSEC** in quantitative results tables. To assess the quality of our reconstruction method in low-light conditions, we evaluate it on the three night driving sequences from the MVSEC dataset using a no-reference metric. This evaluation comprises a total of 9415 reconstructed frames, with an average rate of $10\,\mathrm{Hz}$. The scores from this evaluation are reported under the name **NIGHT** in the quantitative results tables.

High-Quality Frames (HQF) dataset. HQF dataset has 14 indoor and outdoor image sequences that cover a diverse range of motions. The data is captured by two different DAVIS240C cameras with varying noise and contrast threshold characteristics. Both cameras generate events and frames from the same 240×180 pixel array. The camera parameters and scenes are carefully selected to ensure that the ground truth frames are well-exposed and minimally motion-blurred. The dataset provides ground truth intensity frames with an average rate of $22.5\,\mathrm{Hz}$, and following [3], we use the entire sequences for evaluation. In total, we use 15498 ground truth frames for this evaluation. We report these evaluation scores under the name **HQF** in our quantitative results tables. We also consider a specifically curated subset of the HQF dataset to assess the quality of reconstructions under slow motion, which also poses challenges for event-based video reconstruction due

to the reduced event rate. This subset, which we denote as **SLOW** and specifically utilize to analyze the effect of context information, includes all 2333 ground truth frames from two sequences named `desk_slow` and `slow_hand`, which were collected with the explicit aim of incorporating slow-motion scenarios.

UZH-FPV Drone Racing (FPVDR) dataset. This dataset is captured by a miniDAVIS346 (mDAVIS) camera mounted on a quadrotor flown by an expert drone racing pilot with fast and aggressive movements. The dataset consists of 26 indoor and outdoor flight sequences, with a total flight distance of more than $10\,\mathrm{km}$. The events and frames are generated from the same 346×260 pixel array of mDAVIS, which is positioned either forward facing or 45-degree downward facing for each flight. We use this dataset to assess the quality of the reconstructions under fast camera motion, using a no-reference metric. We exclude the first few seconds of each sequence to start quantitative evaluation after the drone takes off. We use event groups that span $40\,\mathrm{ms}$ and evaluate the total 31067 reconstructed frames.

Color Event Camera Dataset (CED). This dataset consists of frames and events collected with a Color-DAVIS346 [8] camera, at 346×260 resolution. We use a few sequences, including simple objects with vibrant colors and scenes with challenging lighting conditions, to present visual results of color reconstructions.

II. Evaluation Metrics

MSE. Mean squared error is a standard metric without parameters. The only thing that can affect the result of MSE while comparing two images is the range of pixel values that images have. We use floating point pixel values in the range [0,1] to calculate MSE. Lower MSE scores are better.

SSIM. For structural similarity, we use the implementation from the scikit-image library [9], v0.19.3. We adjusted the parameters to use the Gaussian weighting scheme described in the original paper [10]. Similar to MSE, we input images with floating point pixel values in the range [0,1] to SSIM calculation. Higher SSIM scores are better.

LPIPS. For LPIPS [11] we use v0.1.4 of the official implementation[1] with pre-trained AlexNet [12] network, which requires normalizing the images so that their pixel values are in the range [-1,1]. Lower LPIPS scores are better.

BRISQUE. For BRISQUE [13], we use the implementation in IQA-PyTorch[2] toolbox [14], v0.1.5, with default settings. The implementation supports 3-channel RGB images; thus, we convert intensity images into RGB images by concatenating three copies of the grayscale image along the third dimension before calculating the scores. The pixel values are again in the range of [0,1]. Lower BRISQUE scores are better.

III. Ablation Study and Further Analysis

In the following ablation studies, we evaluate various design elements of the HyperE2VID model to verify their impact on performance. This includes a detailed comparison against the E2VID+ network [3], which shares similarities with our base network and employs the same training data. Specifically, we retrain E2VID+ with the same hyperparameters as HyperE2VID to assess the influence of these parameters independently of our hypernetwork architecture. We further investigate the role of previous reconstructions by modifying the E2VID+ architecture to include them. Additionally, we compare our context-guided per-pixel dynamic convolutions with standard dynamic convolutions, confirming the superiority of our approach.

A significant part of our ablation study focuses on the use of context information. We experiment with networks using only event voxel grids as context, only previous reconstructions as context, or a combination of both, along with variations in curriculum learning and convolutional context fusion. We then analyze the effect of using different event grouping strategies. Finally, we evaluate the quality of reconstructions in two other challenging scenarios: high frame rate video generation (200 to 5000 FPS) and reconstruction during motionless periods.

Training Settings. We retrained E2VID+ using the same setup and hyperparameters as HyperE2VID to test if E2VID+ could benefit from our hyperparameter choices, without our hypernetwork architecture. The results, shown in the second row of Table I, reveal mixed outcomes. While the retrained E2VID+ shows improvements with respect to the original one in the ECD, FAST, NIGHT, and FPVDR datasets, it falls short in the MVSEC and HQF datasets. This inconsistency suggests that the enhancements are not solely due to optimizing the hyperparameters. A direct comparison with HyperE2VID, under identical conditions, clearly shows the superiority of our hypernetworks-based approach.

Previous Reconstructions. In another experiment, we modify the E2VID+ architecture to include reconstructed intensity image from the previous timestep ($\hat{I}_{k-1}$) along with the current event tensor (V_k) via concatenation at the input. This is to distinguish the benefits of our architectural features from the simple use of past reconstructions. Even with the addition of curriculum learning, similar to HyperE2VID, this variant (shown in the third row of Table I) underperforms compared to both the standard and retrained E2VID+. This highlights the unique effectiveness of our hypernetworks and dynamic per-pixel convolutions.

Dynamic Convolutions. We also compare our context-guided per-pixel dynamic convolutions with standard dynamic convolutions that lack these features. Training networks with dynamic convolutions [15] or CondConv [16] instead of the proposed CGDD block leads to a significant drop in performance, as shown in the fourth and fifth rows of Table I. It highlights the effectiveness of the proposed context-guided per-pixel dynamic convolutions in HyperE2VID in enhancing reconstruction quality.

Context Information. Moreover, we carry out several ablation experiments in order to evaluate the design choices regarding the context information used for guiding the dynamic filter generation process in HyperE2VID. Specifically, we investigate hypernetworks that use only event voxel grids as context,

[1]https://github.com/richzhang/PerceptualSimilarity

TABLE I
RESULTS FROM ABLATION EXPERIMENTS INVESTIGATING EFFECTS OF TRAINING SETTINGS, USE OF PREVIOUS RECONSTRUCTIONS, DYNAMIC CONVOLUTIONS, AND HYPERNETWORKS.

	ECD			MVSEC			HQF			FAST	NIGHT	FPVDR
	MSE	SSIM	LPIPS	MSE	SSIM	LPIPS	MSE	SSIM	LPIPS	BRISQUE		
E2VID+	0.070	0.503	0.236	0.132	0.262	0.514	0.036	**0.533**	**0.252**	22.627	12.285	18.677
E2VID+ (re-trained)	0.047	0.537	0.217	0.153	0.259	0.531	0.048	0.507	0.285	17.719	8.131	15.432
w/ $\hat{I}_{k-1}$ at input	0.077	0.479	0.259	0.226	0.218	0.567	0.037	0.496	0.270	18.830	10.983	21.176
w/ Dynamic Conv. [15]	0.060	0.503	0.246	0.119	0.270	0.493	**0.031**	0.529	**0.252**	15.162	6.602	20.758
w/ CondConv [16]	0.044	0.565	0.221	0.119	0.271	0.504	0.033	0.529	0.254	15.543	6.670	14.264
HyperE2VID	**0.033**	**0.576**	**0.212**	**0.076**	**0.315**	**0.476**	**0.031**	0.530	0.257	**14.024**	**5.973**	**14.178**

TABLE II
ABLATION RESULTS OF HYPERE2VID VARIANTS WHERE WE ALTER THE CONTEXT INFORMATION, THE EXISTENCE OF CONVOLUTIONAL CONTEXT FUSION (CF) BLOCK, AND CURRICULUM LEARNING (CL) STRATEGY.

			ECD			MVSEC			HQF			SLOW			FAST	NIGHT	FPVDR
Context	CL	CF	MSE	SSIM	LPIPS	MSE	SSIM	LPIPS	MSE	SSIM	LPIPS	MSE	SSIM	LPIPS	BRISQUE		
EVG			0.048	0.543	0.219	0.189	0.232	0.549	0.050	0.504	0.280	0.064	0.496	0.333	**13.42**	**1.05**	**4.66**
PR			0.050	0.536	0.229	0.181	0.228	0.573	0.035	0.517	0.276	0.039	0.558	0.283	18.79	10.12	16.18
EVG+PR			0.039	0.559	**0.212**	0.152	0.261	0.532	0.036	0.525	0.271	0.045	0.566	0.279	18.64	9.21	14.44
EVG+PR		✓	0.044	0.548	0.218	0.113	0.274	0.516	0.039	0.520	0.266	0.044	0.569	0.268	17.73	7.38	12.46
EVG+PR	✓		0.038	0.556	0.216	0.120	0.265	0.506	0.032	**0.534**	0.259	0.039	0.541	0.285	18.24	6.09	13.26
EVG+PR	✓	✓	**0.033**	**0.576**	**0.212**	**0.076**	**0.315**	**0.476**	**0.031**	0.530	**0.257**	**0.026**	**0.581**	**0.250**	14.02	5.97	14.18

only previous reconstructions as context, or a combination of both, denoted as EVG, PR, and EVG+PR, respectively. It should be emphasized that these HyperE2VID variants specifically modify the context tensor computation within the CF block, while maintaining the event tensor at the input of the head layer and preserving the dynamic network architecture of both the DFG and CGDD blocks. For EVG+PR, we also examine the impact of using the curriculum learning strategy (CL) and convolutional context fusion (CF). When CF is not used, we concatenate the previously reconstructed images and event tensors channel-wise and downsample the resulting tensor to match the input of the dynamic convolution in the CGDD block. The results are summarized in Table II. Here, we also give the results on the SLOW subset of the HQF dataset, containing slow motion.

Our quantitative results highlight the significance of choosing the right context based on the scene's characteristics. For instance, in slow-motion scenarios (SLOW), the network utilizing solely previous reconstructions (PR) vastly outperforms the one using only event voxel grids as context (EVG). Conversely, in scenes with fast motion (FAST, FPVDR) or low light conditions (NIGHT), PR's performance diminishes. To visually illustrate these findings, Fig. 1 shows two representative scenes from our test datasets. The first scene, from the FAST segment of the ECD dataset, highlights the limitations of standard camera intensity frames for fast motion, which struggle with either motion blur or underexposure, while the event data adeptly captures the dynamic edges of the scene. This effectively demonstrates the strength of event data in high-speed conditions. The second scene, from the SLOW segment of the HQF dataset, presents a slow-motion environment where intensity frames capture detailed visual information, but events are generated sparsely, capturing only significant brightness changes. Consequently, much of the visual detail in the scene is not visible in the event data.

Our findings in Table II also reveal that leveraging both events and previous reconstructions as contextual information (EVG+PR) generally outperforms using only events (EVG) or only reconstructions (PR) as context. When using both events and reconstructions (EVG+PR), incorporating only the context fusion (CF) yields performance improvements on the MVSEC dataset. In contrast, incorporating only the curriculum learning strategy (CL) enhances performance on both the MVSEC and HQF datasets. Combining all these components results in our proposed HyperE2VID model (last row), which achieves the best scores on ECD, MVSEC, and HQF datasets, and second-best scores on FAST and NIGHT datasets. The variant using only event voxel grids as context (EVG), despite struggling on the HQF dataset and especially in its slow-motion sequences, excels in the fast-motion and night driving sequences. This is also visible in the top row of Fig. 1, where the reconstruction of EVG is sharper and has minimal artifacts, even compared to the reconstruction of HyperE2VID. While HyperE2VID achieves the highest scores overall, the superior performance of the event-only model in certain scenarios suggests potential room for improvement for our context fusion block for future work.

Event Grouping. We perform additional experiments to assess the effect of using different event grouping strategies, that is,

forming event voxel grids with different temporal windows and event numbers. First, we investigate the case with fixed temporal window grouping and conduct ten sets of experiments, each utilizing a different temporal window ranging from 10 ms to 100 ms. Second, we examine the case of fixed number event grouping and perform ten additional sets of experiment runs, each employing fixed number event grouping with a different event count ranging from 2K to 45K. For each experiment set, we consider the four best-performing methods (E2VID+, FireNet+, ET-Net, and HyperE2VID) and reconstruct videos with them using events from ECD, MVSEC, and HQF datasets by utilizing the selected event grouping strategy. For all these experiments, we employ a tolerance of 1 ms to match the reconstructions with ground truth frames and calculate LPIPS scores whenever there is a match. We then compute mean LPIPS scores for each method and for each experiment. The results, given in Fig. 2, demonstrate the superiority of the proposed HyperE2VID architecture for generating high-quality reconstructions over a wide range of event grouping settings.

High Frame Rate Video Reconstruction. For high frame rate video reconstruction, Rebecq *et al*. [7] suggested a method that groups a fixed number of events and runs multiple reconstructions in parallel, each with a slight temporal shift. This technique, however, necessitates the selection of an event count and a temporal shift value. This involves conducting numerous separate reconstructions to produce a set of videos, which are then merged by reordering frames and subjected to temporal filtering to mitigate flickering, ultimately yielding a video with a variable frame rate. In contrast, our approach utilizes fixed-temporal-window event grouping without the need for temporal shifts or parallel reconstructions, facilitating the generation of a high and constant frame rate video. The temporal window is straightforwardly determined based on the desired frame rate, using the formula 1/FPS, where a smaller window correlates with a higher FPS. This simplistic method reveals that most event-based video reconstruction networks from existing literature begin to falter in visual quality when the FPS exceeds one thousand, as the event voxel grid statistics start to diverge from the conditions they were trained under. HyperE2VID, however, consistently produces high-contrast, sharp reconstructions, even at frame rates of several thousand frames per second. Fig. 3 presents reconstructed videos at high frame rates, ranging from 200 FPS to 5000 FPS. Owing to its dynamic network architecture, HyperE2VID adeptly adjusts to the varying event statistics, thus maintaining superior visual quality in high FPS video output.

Reconstruction During Still Periods. Another challenging case for event-based video reconstruction is the stationary sections in event sequences since the event rate drastically reduces, with only noise events being generated by the camera. Here, we qualitatively analyze the reconstruction quality of HyperE2VID and other methods during these motionless periods by presenting their reconstructions in Fig. 4. The desired functionality for methods is to retain their most recent reconstructions during the pause segment, but most of them start to generate intensity images with degraded quality within a few seconds. On the other hand, the results presented in the last row of Fig. 4 demonstrate HyperE2VID's ability to preserve its high contrast and sharp reconstructions during the motionless segments, thanks to its dynamic network architecture, allowing it to adapt to highly varying event data.

IV. Post-Processing

Here, we describe a post-processing procedure and present its qualitative results, which can optionally be applied to reconstructions of any event-based video reconstruction method, eliminating or minimizing various types of artifacts that might be encountered in textureless regions.

Our post-processing procedure consists of three steps: i) obtaining a filtered version of the reconstruction, ii) obtaining a soft mask to segment textureless regions from other regions, and iii) blending the original reconstruction with the filtered version of it using the soft mask. In the first step, we aim to use a simple filter that can remove artifacts in textureless regions but can also degrade image quality in other regions. We use a simple median filter with a kernel size of 3×3 for this, but one can use other filters according to the type of artifacts targeted. In the second step, we aim to obtain a soft mask that can segment textureless scene regions. For this, we choose to accumulate incoming events over a 2D image and decay them exponentially with time to give less weight to events further in the past, similar to a time-surface [17]. Since

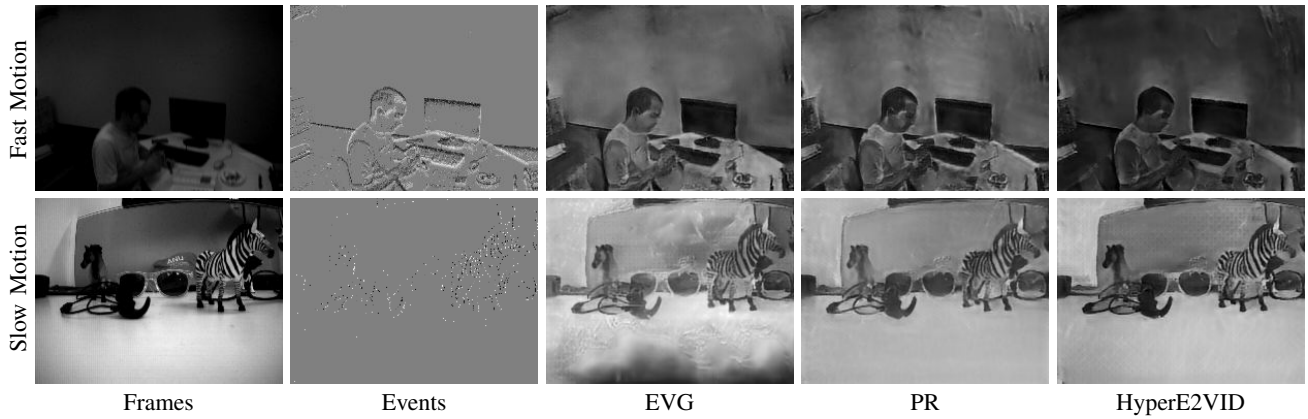
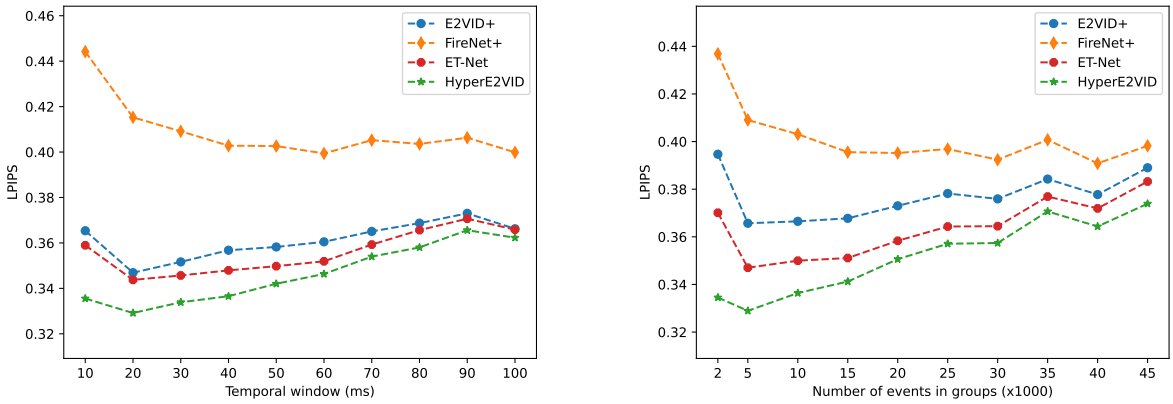


Fig. 1. **Understanding the role of context information.** This figure shows frames, events, and reconstructions from two distinct scenes: one with fast motion (top) and another with slow motion (bottom). It highlights the significance of utilizing event and reconstruction data as context information for optimal results.



(a) Different temporal windows

(b) Different event numbers

Fig. 2. **Effect of using event voxel grids with different temporal windows and event numbers.** We consider four best performing methods (E2VID+, FireNet+, ET-Net, and HyperE2VID), and compute their mean LPIPS scores obtained on ECD, MVSEC, and HQF datasets, using a variety of event grouping settings. (a) We conduct ten sets of experiments, each using a different temporal window ranging from 10 ms to 100 ms. (b) We conduct ten experiment runs, each utilizing fixed-number event grouping with a different event count ranging from 2K to 45K. For (a) and (b), we employ a tolerance of 1 ms to match the reconstructions with ground truth frames, and calculate LPIPS scores whenever there is a match. Then, we plot mean LPIPS scores across these experiments runs for each method. The results demonstrate the superiority of the proposed HyperE2VID architecture for generating high-quality reconstructions, over a wide range of event grouping settings.

events are mostly generated from textured regions, this event image gives us a good approximation to segment textureless areas. We then apply dilation and Gaussian blur with 5×5 kernels to this event image, to obtain the final soft mask for that time step. We then use this mask to simply blend the original reconstruction with the filtered version of it, giving more weight to the latter for textureless regions.

The qualitative results of this procedure are presented in Fig. 5. We consider reconstructions of three models, E2VID+, ET-Net, and HyperE2VID, and show the effect of applying post-processing on them, using two scenes from the ECD and HQF datasets. Here, it can be seen that the described procedure generates visually pleasing images, by removing or minimizing most of the artifacts, such as checkerboard patterns. While it effectively removes fine-scale artifacts, the larger-scale artifacts remain, such as the ones in ET-Net's reconstruction in desk_slow sequence, since we only employ a simple median filter with a small kernel size of 3×3 pixels. Although simple, the presented post-processing procedure can improve the visual results of event-based video reconstruction models in certain scenarios, indicating the potential for improvements in future work.

REFERENCES

[1] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.

[2] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, 2018.

[3] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, "Reducing the sim-to-real gap for event cameras," in *ECCV*, 2020.

[4] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *ICRA*, 2019, pp. 6713–6719.

[5] C. Scheerlinck, H. Rebecq, T. Stoffregen, N. Barnes, R. Mahony, and D. Scaramuzza, "CED: color event camera dataset," in *CVPRW*, 2019.

[6] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[7] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[8] G. Taverni, D. P. Moeys, C. Li, C. Cavaco, V. Motsnyi, D. S. S. Bello, and T. Delbruck, "Front and back illuminated dynamic and active pixel vision sensors comparison," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 5, pp. 677–681, 2018.

[9] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[11] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012.

[14] C. Chen and J. Mo, "IQA-PyTorch: Pytorch toolbox for image quality assessment," [Online]. Available: https://github.com/chaofengc/IQA-PyTorch, 2022.

[15] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *CVPR*, 2020, pp. 11 030–11 039.

[16] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," *NeurIPS*, vol. 32, 2019.

[17] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "Hots: a hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, 2016.

Fig. 3. **High frame rate video synthesis.** We employ a simple approach with fixed-temporal-window event grouping for generating videos with high FPS. Here we present frames corresponding to the first second of the `slider_depth` sequence from the ECD dataset, taken from videos reconstructed at 200 Hz, 500 Hz, 1 kHz, 2 kHz, and 5 kHz, which are generated by using temporal windows of 5 ms, 2 ms, 1 ms, 500 µs, and 200 µs, respectively. While most of the other methods start to generate videos with lower visual quality as we increase FPS above one thousand, HyperE2VID maintains its high contrast and sharp reconstructions even when generating videos with several thousand frames per second.

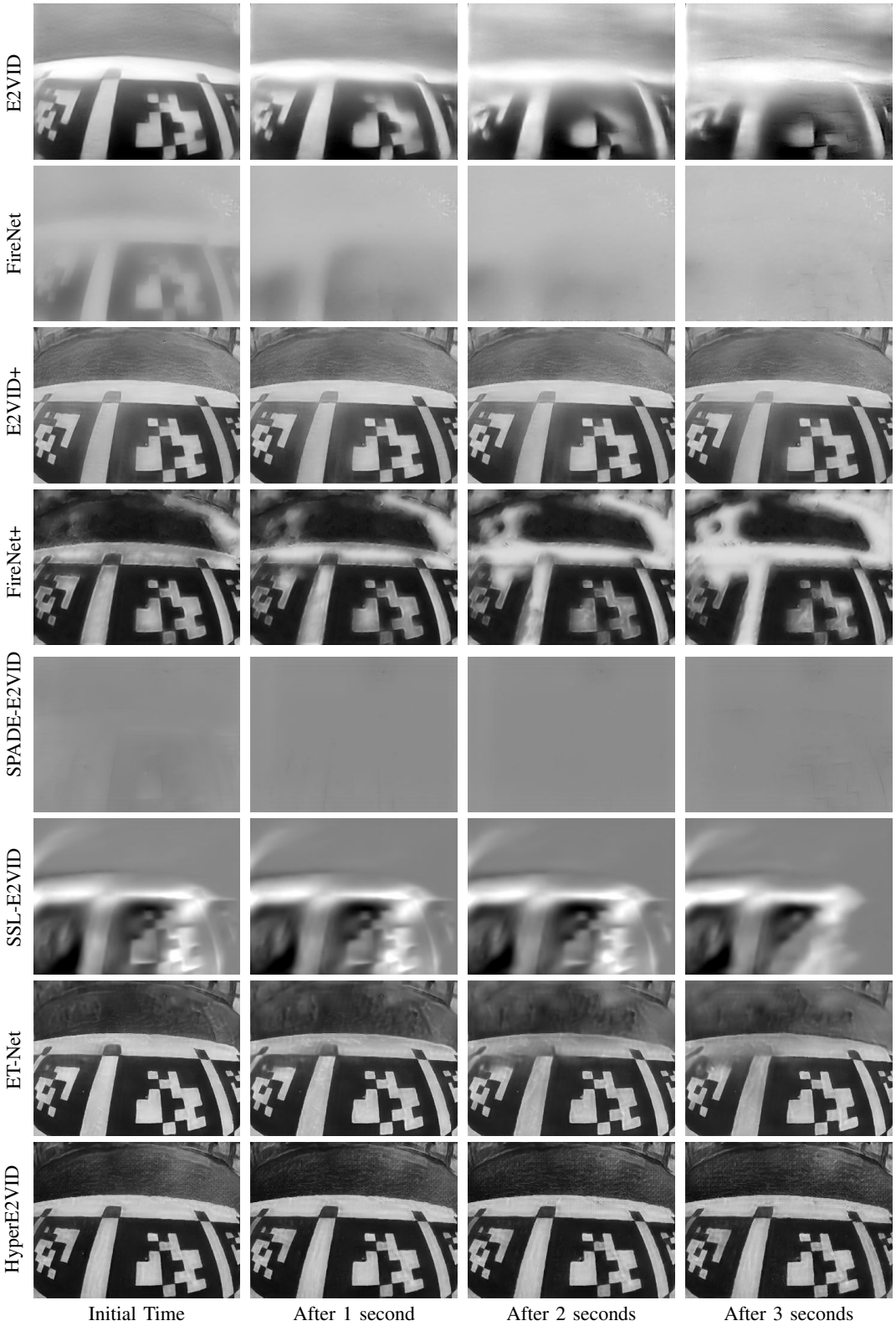


Fig. 4. **Assessing reconstruction quality in motionless sections.** Stationary sections in event sequences pose additional challenges for video reconstruction since the event rate drop-offs to almost zero, with only noise events being generated. Here, we consider a segment from the UZH-FPV Drone Racing dataset, where the drone lands on a board with ArUco markers and stops. For each method, we present reconstructions from the initial time just after the drone stops in the leftmost column and three more reconstructions at one-second intervals in subsequent columns. The desired functionality for methods is to retain their most recent reconstructions during the pause segment, but most of them start to generate intensity images with degraded quality within a few seconds by gradually decaying images and revealing artifacts such as blurry and bleeding edges. On the other hand, HyperE2VID manages to preserve its high contrast and sharp reconstructions during the motionless segments, thanks to its network architecture, which allows it to dynamically adapt to highly varying event data.

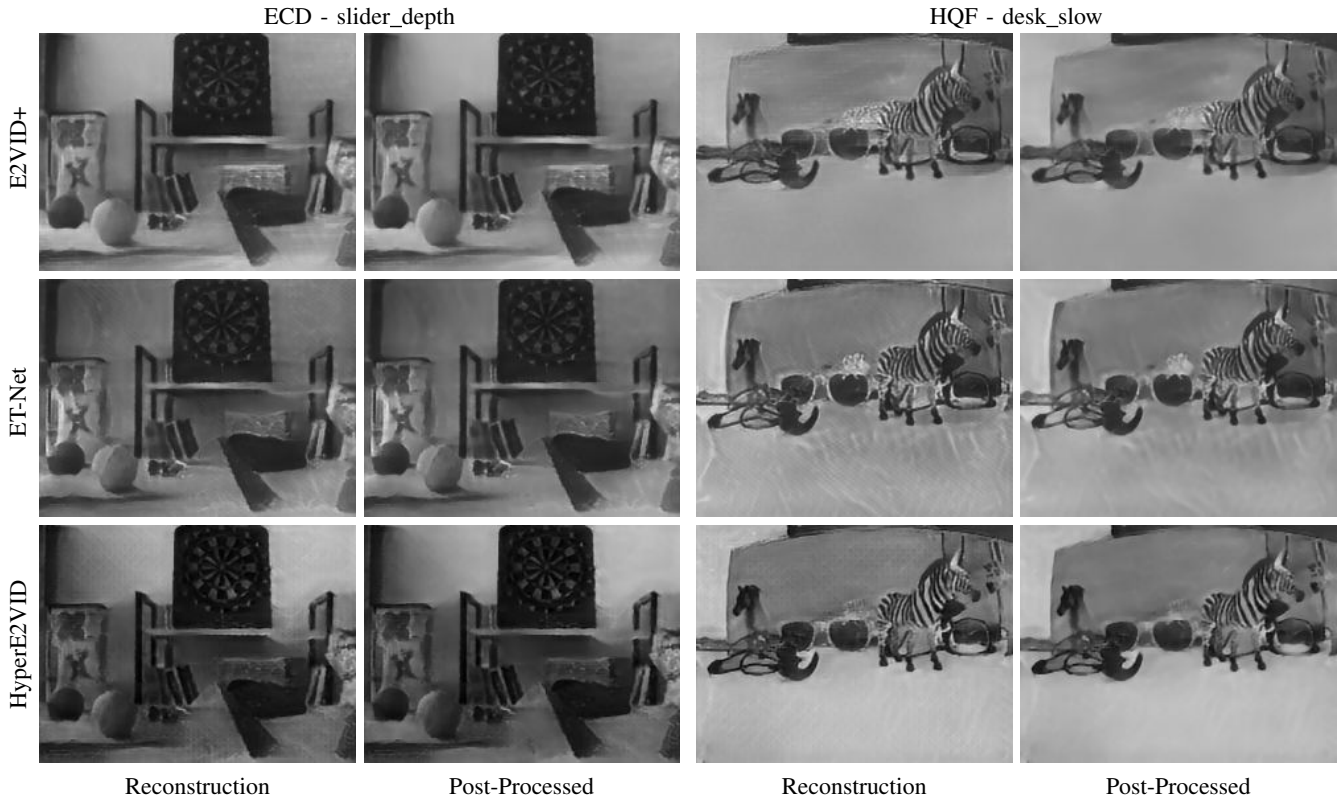


Fig. 5. **Visual results of post-processing.** Here, we consider two scenes from the ECD and HQF datasets and present reconstructions of E2VID+, ET-Net, and HyperE2VID for each scene, with or without post-processing. The results demonstrate that the post-processing can satisfactorily remove or minimize most of the fine-scale artifacts, such as checkerboard patterns.