Bridging the Gap between Events and Frames through Unsupervised Domain Adaptation

Nico Messikommer, Daniel Gehrig, Mathias Gehrig, Davide Scaramuzza

***Abstract*—Reliable perception during fast motion maneuvers or in high dynamic range environments is crucial for robotic systems. Since event cameras are robust to these challenging conditions, they have great potential to increase the reliability of robot vision. However, event-based vision has been held back by the shortage of labeled datasets due to the novelty of event cameras. To overcome this drawback, we propose a task transfer method to train models directly with labeled images and unlabeled event data. Compared to previous approaches, (i) our method transfers from single images to events instead of high frame rate videos, and (ii) does not rely on paired sensor data. To achieve this, we leverage the generative event model to split event features into content and motion features. This split enables efficient matching between latent spaces for events and images, which is crucial for successful task transfer. Thus, our approach unlocks the vast amount of existing image datasets for the training of event-based neural networks. Our task transfer method consistently outperforms methods targeting Unsupervised Domain Adaptation for object detection by 0.26 mAP (increase by 93%) and classification by 2.7% accuracy.**

***Index Terms*—Deep Learning for Visual Perception, Object Detection, Segmentation and Categorization, Transfer Learning**

Multimedia Material

The code of this project is available at https://github.com/uzh-rpg/rpg_ev-transfer Additional qualitative results can be viewed in this video: https://youtu.be/fZnBSqni6PY

I. Introduction

THE outstanding properties such as high dynamic range, high temporal resolution, and low latency make event cameras promising for several robotic and automotive applications in edge-case scenarios, such as high dynamic range and fast relative motion. However, event cameras suffer from a recurring issue typical of any novel sensor modalities: the lack of labeled datasets. Event-based datasets represent only 3.14% of the existing vision dataset [1], [2].

Instead of capturing images at a fixed rate, event cameras measure changes of intensity asynchronously per pixel. This results in a stream of events that encodes the time, location, and polarity of the intensity change. For a more in-depth survey, we refer to [3]. Despite the radical different working principle, the output of event and frame-based cameras still contains a significant information overlap, as both cameras share the underlying principle of capturing the scene irradiance through an optical system [4].

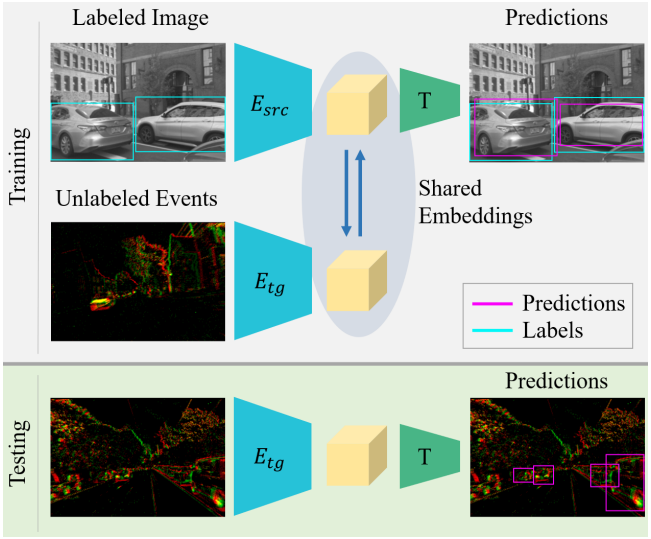

Fig. 1: Our approach can teach a network to detect cars in event frames even though it was never told how cars look in the event domain. This unsupervised domain adaption is possible by leveraging labeled grayscale images and unlabeled events. During testing, our approach consists of a simple encoder E and task network T and thus has no computational overhead of first translating events to images.

In this work, we show how this information overlap can be leveraged for *Unsupervised Domain Adaptation (UDA)* of event-based networks, in which labeled source and unlabeled target data are available to transfer a task to the target domain. Previous work has considered task transfer either between *paired* events and frames, *i.e.* recorded on the same pixel array [2], [5], or through video-to-event translation [6], [7]. These settings provide strong supervision between modalities, by providing per-pixel constraints or direct motion information from video, but severely limit the kinds of datasets which can be used. In fact, most large-scale datasets only comprise *still images* [8], [9], [10] instead of video, and were not recorded with a colocated event sensor, and are thus unpaired. Here, we present a method that can directly leverage these datasets, and does not rely on paired data or video inputs, opening up the immense bulk of frame-based datasets for event-based vision.

Furthermore, our approach does not rely on hand-designed generative models for video-to-events generation [7], [6], [11] and has no computational overhead during inference by converting events to intensity images [4].

To bridge the gap between frames and events without paired

Manuscript received: September, 9, 2021; Revised December, 7, 2021; Accepted January, 3, 2022.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation (SNSF).

The authors are with the Robotics and Perception Group, Department of Informatics, University of Zurich, and Department of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland (http://rpg.ifi.uzh.ch. nmessi@ifi.uzh.ch).

Digital Object Identifier (DOI): see top of this page.

data, we introduce a novel *single-image-to-event* translation technique based on the event generation model combined with standard image-to-image translation techniques [12], [13], [14]. Crucially, instead of learning image-to-event translation directly, it only learns to correct initial guesses from the generative model. However, event generation from a single image is ill-posed since motion information is missing. We solve this by explicitly extracting motion features from events in addition to the shared features that contain domain invariant information about the scene. To achieve this split, we introduce a shared embedding discriminator and enforce shared feature consistency using sensor-specific knowledge. Our approach can successfully transfer from images to events by leveraging large-scale datasets, which we show by outperforming state-of-the-art methods targeting UDA, for both object detection in MVSEC [15] and classification on N-Caltech101 [16].

Our contributions can be summarized as follows.

1) We propose a transfer learning method that uses labeled frame-based datasets together with unlabeled events recorded in a target environment to train event-based networks. In particular, we show that networks trained on labeled daylight images can be transferred to challenging nighttime scenarios, where event cameras outperform standard cameras thanks to their higher dynamic range.
2) Our approach leverages prior sensor knowledge based on the event generation model, and introduces a mapping from events to motion and content embeddings. This opens existing image-based datasets for event cameras, by transferring single images to events.
3) Neural networks trained by our task transfer method outperform state-of-the-art object detection methods which target UDA by 0.27 mAP (an increase of 93%) and achieve a 2.7% accuracy increase in the classification task outperforming even some supervised approaches.

II. Related Work

Unsupervised Domain Adaptation The general problem of transfer learning based on labeled source data and unlabeled target data has accumulated vast literature over the years. For a survey, we refer the reader to [17]. Early deep-learning-based methods use discriminators [18] or gradient reversal layers [19] to align the embedding space of source and target domain samples.

Analogously, *image-to-image translation* methods can be used to induce shared embedding spaces by mapping samples from target to source domain, or vice versa. While the former enables the reuse of pre-trained networks trained in the source domain [13], the latter can provide a source of labeled datasets in the target domain by converting datasets from the source domain [20], [21].

To directly transfer a task, recent work [22], [23] proposed to jointly learn the task and mapping from input to domain-shared and domain-invariant features. However, they can mostly work with shared network layers because they consider a smaller domain gap. The domain gap between sensor modalities (RGB vs. Infrared) can also be exploited to supervise a RGB detection network in night sequences [24].

One work that addresses a large domain gap in the context of biomedical imaging is [25], where domain-invariant features are constructed from source and target characteristics. Similarly, our task-transfer approach addresses the large domain gap between two complementary but different vision sensors: standard cameras and events cameras.

Event-based Approaches To address the lack of labeled datasets, a recent class of methods seeks to convert events to high dynamic range (HDR) image reconstruction through supervised [4] or adversarial training [26], [27], [28]. With these images, standard pre-trained neural networks trained on images can be used. However, despite this advantage, these methods impose a computational overhead by first generating image reconstructions. Instead, [2] simply adapt the first few layers of pre-trained frame-based networks to event data by enforcing feature consistency between the two separate sensor encoders. While this eliminates the need for costly event preprocessing, it requires paired images and events, *i.e.* events and images recorded on the same sensor and scene to adapt a given network. By contrast, the method in [29] is designed to work with unpaired data but only converts between events in different illumination conditions. The first works to leverage existing frame-based datasets were *video-to-event translation* methods. These methods either rely on model-based [11], [4], [6], [7] or data-driven [5] approaches to convert video sequences into artificial events, which can be used to directly train neural networks on event data. This opened up the possibility of training networks for event data on larger and more diverse datasets. However, these methods are still limited to translating video to events, thus ignoring the majority of existing datasets that comprise images. The work most similar to ours leverages affinity graphs to perform the task transfer from frames to events [30]. In comparison, our approach splits the embedding space into shared and sensor-specific features and leverages the event generation model to align both domains.

In this work, we introduce a novel method that addresses the limitations of previous approaches by performing unsupervised domain adaptation, which *(i)* maps unpaired images and events to a shared embedding space, *(ii)* leverages single-image-to-event instead of video-to-event translation, and *(iii)* performs task-transfer by jointly training a task-specific network on the shared embedding. We introduce a novel single-image-to-event translation module that combines the event generation model [31] with standard translation methods. Moreover, our method maps event data into separate content and sensor-specific features and only matches content features across modalities. In doing so, we take inspiration from style-transfer techniques [32], [12], [13].

III. Method

Our goal is to train a network on labeled images for a specific task and transfer the network to events, such that the network successfully performs the task in the event-domain without requiring any labeled events nor paired images and event data, see Fig. 1. This setting of transferring a task from a labeled source domain (image domain Y_{img}) to an

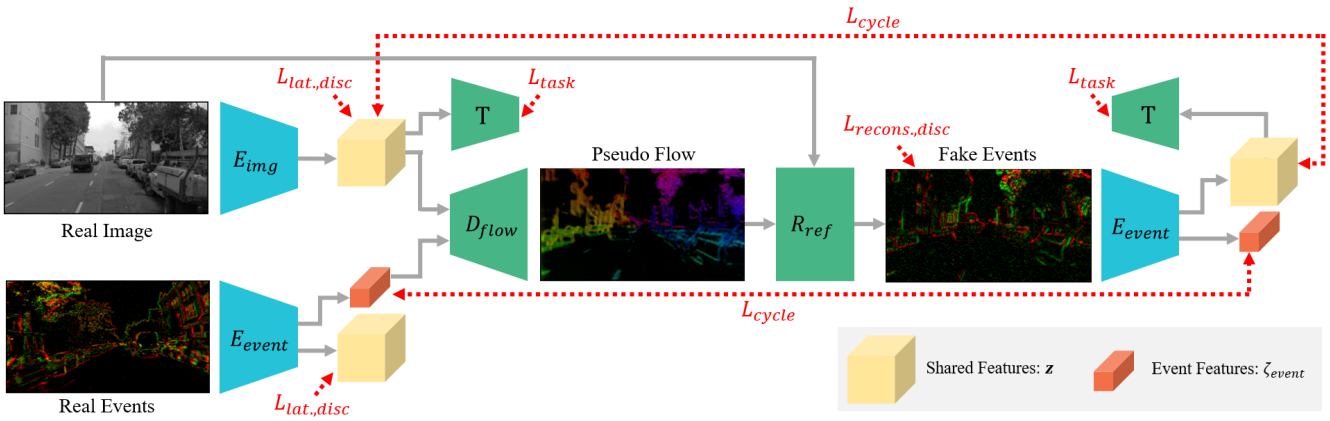

Fig. 2: During training, our method uses single-image-to-event translation to transfer a task from the image to the event domain. As there is a large domain gap between events and grayscale images, we use two separate encoders E_{img} and E_{event} (blue) to process unpaired images and event frames. Shared features $\mathbf{z}$ (yellow) and event-specific features ζ_{event} (orange) are extracted from the event frame. Both features are given as input to the event decoder D_{flow}, which creates a pseudo flow map. This flow map is combined with the image to create clean events using the event generation model. To model sensor noise, the refinement module takes additionally random input feature maps. The task network T takes shared features from the images and the fake events as input and generates task predictions, which are supervised with the image labels. The applied loss constraints are visualized with red arrows. During inference, only the event encoder E_{event} and the task network T are required, both are computationally light-weight networks.

unlabeled target domain (event domain Y_{event}) is generally defined as Unsupervised Domain Adaptation, short UDA. The task transfer is possible since event and frame-based cameras share the underlying principle of capturing the scene irradiance through an optical system. Therefore, an information overlap exists on which the task can be learned on images and transferred to events.

In Sec. III-A, we present the general network architecture and the latent space split into shared and sensor-specific features, which facilitates the task transfer. The alignment of the shared latent space is enforced through multiple losses, which are explained in Sec. III-B. As a common constraint in the UDA literature [33], [34], [35], [22], [25], [13], we perform domain translation to generate pseudo pairs, which are used to align the shared latent space on which the task is learned. However, compared to classical UDA approaches, we only use a one-sided translation from images to events to achieve a better embedding alignment. In Sec. III-D, we introduce our novel event construction from a single image based on the event generation model [31], which is explained in Sec. III-C. Since the event generation model constitutes a relation between events, image gradients and optical flow, we can strongly constraint the image-to-event translation and thus significantly improve the task transfer. It is important to state that the image-to-event translation is only applied as an auxiliary task to help to transfer the task from images to events. As we directly optimize the task transfer from images to events, our method consistently outperforms pure translation methods [6], [36], [4], [5].

A. General Model Architecture

In our framework, events $\mathbf{y}_{\text{event}}$ and images $\mathbf{y}_{\text{img}}$ are processed with separate encoders E_{img} and E_{event} due to the large domain gap between Y_{img} and Y_{event}, as shown in Fig. 2. As the asynchronous output signal of event cameras also contains motion information, event cameras measure specific features ζ_{event} about the scene, which standard cameras can not perceive in a single frame. This non-overlapping information, however, hinders the image-to-event task transfer as it is impossible to fully align the embedding space. We solve this by separating event features into *sensor specific* features ζ_{event}, which contain motion information, and *content* features $\mathbf{z}_{\text{event}}$, which carry information shared in both domains Y_{img} and Y_{event}.

$$\begin{aligned} \mathbf{z}_{\text{img}} &= E_{\text{img}}(\mathbf{y}_{\text{img}}) \\ \mathbf{z}_{\text{event}} = E_{\text{event}}(\mathbf{y}_{\text{event}}) \qquad & \zeta_{\text{event}} = E_{\text{event, attr}}(\mathbf{y}_{\text{event}}). \end{aligned} \tag{1}$$

The resulting shared features $\mathbf{z}_{\text{img}}$ and $\mathbf{z}_{\text{event}}$ are given as input to the task branch T, which computes the task-specific output. To generate pseudo event and image pairs, shared features from an image $\mathbf{z}_{\text{img}}$ are combined with event-specific features ζ_{event} from a random event sample to compute a pseudo-flow field using a flow decoder D_{flow}. The resulting pseudo-flow and the input image are then converted to events $\hat{\mathbf{y}}_{\text{event}}$ in the refinement network R_{ref},

$$\hat{\mathbf{y}}_{\text{event}} = R_{\text{ref}}(D_{\text{flow}}(\mathbf{z}_{\text{img}}, \zeta_{\text{event}})). \tag{2}$$

Single-image-to-event translation is explained in more detail in Sec. III-D. The overall architecture is depicted in Fig. 2.

B. Shared Latent Space Constraints

The unsupervised task transfer from images to events requires multiple constraints as there is neither task supervision in the event domain nor paired sensor data. Therefore, multiple losses are applied to enforce a shared latent space of $\mathbf{z}_{\text{img}}$ and $\mathbf{z}_{\text{event}}$, which ensures that the task branch T successfully performs the task in both domains. As a first constraint, we apply adversarial training [37] with a PatchGAN discriminator network F_{lat} [38] to the latent features $\mathbf{z}_{\text{img}}$ and $\mathbf{z}_{\text{event}}$

$$\begin{aligned} \mathcal{L}_{\text{lat.,disc.}} =& \mathbb{E}_{\mathbf{y}_{\text{img}}}[\max(0, 1 - F_{\text{lat}}(\mathbf{z}_{\text{img}}))] \\ &+ \mathbb{E}_{\mathbf{y}_{\text{event}}}[\max(0, 1 + F_{\text{lat}}(\mathbf{z}_{\text{event}}))] \\ \mathcal{L}_{\text{lat.,gen.}} =& \mathbb{E}_{\mathbf{z}_{\text{img}}}[F_{\text{lat}}(\mathbf{z}_{\text{img}})] - \mathbb{E}_{\mathbf{y}_{\text{event}}}[F_{\text{lat}}(\mathbf{z}_{\text{event}})]. \end{aligned} \tag{3}$$

Similar to [5], we use a hinge-loss [39] and optimize the above loss functions in an alternating fashion. The above objective forces the latent space to be indistinguishable to the

discriminator F_{lat}, and thus, the latent space becomes aligned. As a consequence, the motion information required for the event generation can only be propagated through the event-specific features, which enforces the embedding split.

As an additional constraint, we generate pseudo sensor pairs using a one-sided translation from single images to events. These pseudo-pairs are used to formulate consistency losses on the latent variables $\mathbf{z}_{\text{img}}$ and ζ_{event}, as summarized below:

$$\begin{aligned}\mathcal{L}_{\text{cycle}} = &|\mathbf{z}_{\text{img}} - E_{\text{event}}(R_{\text{ref}}(D_{\text{flow}}(\mathbf{z}_{\text{img}}, \zeta_{\text{event}})))|_1 \\ &+ |\zeta_{\text{event}} - E_{\text{event, attr}}(R_{\text{ref}}(D_{\text{flow}}(\mathbf{z}_{\text{img}}, \zeta_{\text{event}})))|_1\end{aligned} \tag{4}$$

To generate realistic events from a single image, the following adversarial loss is applied on the reconstructed events $\hat{\mathbf{y}}_{\text{event}}$ using an event discriminator F_{event}

$$\begin{aligned}\mathcal{L}_{\text{recons.,disc.}} =&\mathbb{E}_{\hat{\mathbf{y}}_{\text{event}}}[\max(0, 1 - F_{\text{event}}(\hat{\mathbf{y}}_{\text{event}}))] \\ &\mathbb{E}_{\mathbf{y}_{\text{event}}}[\max(0, 1 + F_{\text{event}}(\mathbf{y}_{\text{event}}))] \\ \mathcal{L}_{\text{recons.,gen.}} =&\mathbb{E}_{\hat{\mathbf{y}}_{\text{event}}}[F_{\text{event}}(\hat{\mathbf{y}}_{\text{event}})]\end{aligned} \tag{5}$$

The used constraints are visualized with red arrows in Fig. 2. Overall, these general UDA methods represent a solid basis for closing the domain gap between events and images. However, as shown in our experiments in Sec. IV, current UDA methods fall short in transferring task knowledge between the large gap of events and images. The next section shows how the event generation model can be leveraged to improve the task transfer between the sensor domains.

C. Event Generation Model

The underlying principle of an event and frame camera can be exploited to guide the single-image-to-event translation. As discussed in the introduction of Sec III, event and frame cameras are both optical sensors, which capture the scene irradiance through lenses. Due to this shared principle of measuring the light intensity, images can approximately be translated to events through the theoretical concept of the event generation model [31]. This generative model describes the behavior of an ideal event camera under the assumption of constant brightness and of small time differences Δt. In Eq. 6, $\tilde{I}_{(x,y)} = \log(I_{(x,y)})$ expresses the measured intensity in log space and α represents the angle between the optical flow vector $\mathbf{v}_{(x,y)}$ and image gradient $\nabla\tilde{I}_{(x,y)}$.

$$\begin{aligned}\Delta\tilde{I}_{(x,y)} &\approx -\langle\nabla\tilde{I}_{(x,y)}, \mathbf{v}_{(x,y)}\Delta t\rangle \\ &= -|\nabla\tilde{I}_{(x,y)}||\mathbf{v}_{(x,y)}|\Delta t\cos\alpha\end{aligned} \tag{6}$$

An event is triggered if the log intensity change $\Delta\tilde{I}$ is above a predefined contrast threshold C. Thus, the number N of events at a pixel (x, y) can be approximated according to Eq. 7.

$$N_{(x,y)} \approx \lfloor\Delta\tilde{I}_{(x,y)}/C\rfloor \tag{7}$$

The event generation model enables the transfer from a single image to events, if motion information in form of optical flow $\mathbf{v}_{(x,y)}$ and time difference Δt is provided. By considering only single frames, which most frame-based datasets consist of, the event generation is an ill-posed problem due to the missing motion information. To account for that, we split the events $\mathbf{y}_{\text{event}}$ into two domains: latent space $\mathbf{z}_{\text{event}}$ shared with image features and an additional sensor-specific space ζ_{event}, in which the motion information in the events $\mathbf{y}_{\text{event}}$ is stored. Thus, we can reconstruct artificial events from frames by combining content and sensor-specific features.

D. Event Generation based on Pseudo-Flow

Following Eq. 7, we observe that the predicted events relate to the image gradient $\nabla\tilde{I}_{(x,y)}$ via optical flow. Instead of optical flow, we propose to directly predict pseudo-flow vectors $\hat{\mathbf{v}}_{(x,y)}$, which implicitly contain the unknown parameters Δt, $\cos\alpha$ and C. Thus, we do not need to compute these parameters explicitly.

$$\begin{aligned}\hat{\mathbf{v}}_{(x,y)} &= \mathbf{v}_{(x,y)}\frac{1}{C}\Delta t\cos\alpha \\ \hat{N}_{(x,y)} &\approx \langle\Delta\tilde{I}_{(x,y)}, \hat{\mathbf{v}}_{(x,y)}\rangle\end{aligned} \tag{8}$$

It can be observed in Eq. 8 that the number of predicted events $\hat{N}_{(x,y)}$ can either be changed by the pseudo-flow magnitude or by the angle between the two vectors $\Delta\tilde{I}_{(x,y)}$ and $\hat{\mathbf{v}}_{(x,y)}$. Thus, our pseudo-flow is not equivalent to optical flow as the adversarial training only enforces realistic events by either adjusting the direction or the magnitude of $\hat{\mathbf{v}}_{(x,y)}$.

The pseudo-flow is constructed from a combination of *sensor-specific* and *shared* features. The resulting pseudo-flow field adheres to the content extracted from an image $\mathbf{z}_{\text{img}}$ but with the general motion information of the event data, encoded in the *sensor-specific* feature ζ_{event}. Thus, the pseudo-flow generation also ensures that the *sensor-specific* features are "flow-like" since they should produce realistic events, which is enforced by a discriminator. The event generation based on pseudo-flow constrains the image-to-event translation and supports the adversarial training since it provides good event predictions early during training.

We propose a novel refinement block R_{ref}, which computes the inner product of the predicted pseudo-flow and image gradients as in Eq. (8) to obtain an initial guess of the translated events. The refinement net uses three convolutional layers to predict residual event representations, which are added to the initial guess. These residual events correct for overlapping polarity regions and event noise in the initial reconstruction. The event decoder and refinement block are mainly supervised by the discriminator loss $\mathcal{L}_{\text{recons.,disc.}}$. By enforcing realistic events, the supervision signal is back propagated to the event decoder since the fake events are generated based on the pseudo-flow and image gradient.

E. Pseudo-Flow Augmentation

As the target domain is mostly known, we can augment the event generation by adding an artificial flow field according to the motion present in the target event data. This augmentation loss $\mathcal{L}_{\text{augm}}$ is crucial to enforce the split into sensor-shared and *sensor-specific* features. It is essential to include augmentation consistent with the target event domain. Otherwise, the discriminator easily distinguishes between the translated and real events, thus degenerating the adversarial training and the task transfer. The augmented pseudo-flow field consists of vectors with a magnitude of the pseudo-flow predictions and the directions of the target-specific motion distributions.

Events are then generated in the refinement module R_{ref} from this augmented flow. These events are then processed again by the event encoder E_{event} to obtain event-specific features ζ_{event}, which are combined with the original shared feature to construct new events. This event representation should be identical to the events obtained by the augmented flow as only the event-specific features ζ_{event} contain motion information. Thus, an L^1-loss is applied between those two event representation, as visualized in Fig. 3. The augmentation of reconstructed events can only be propagated through the event-specific features to reconstruct the augmented event representation since the content features stay constant, which further enforces the split into sensor-shared and sensor-specific features. Moreover, the proposed flow augmentation can only be observed in the event representation, which means that the motion-specific attributes have to be extracted from an event sample. Thus, the augmentation loss would not be feasible if there is no unpaired event data. The augmentation loss also helps preventing mode collapse of the event generation since it forces the event decoder to predict a large distribution of flow maps.

F. Summary Loss Constraints

In addition to the above introduced loss constraints, the task loss $\mathcal{L}_{task}$ is applied on the images and the fake events, which both have corresponding image labels. Eq. 9 shows a summary of the presented loss functions.

$$\begin{aligned} L_{Gen} =& \mathcal{L}_{\text{lat,gen.}} + \mathcal{L}_{\text{recons.,gen.}} + \mathcal{L}_{\text{cycle}} + 2\mathcal{L}_{\text{augm}} + \mathcal{L}_{task} \\ L_{Dis} =& \mathcal{L}_{\text{lat,disc.}} + \mathcal{L}_{\text{recons.,disc.}} \end{aligned} \tag{9}$$

IV. EXPERIMENTS

A. Classification

Experimental Setup We validate our approach for event classification on the Neuromorphic-Caltech101 (N-Caltech101) [16] dataset. Since Caltech101 [40] samples were recorded for N-Caltech101, it is a straightforward choice to use Caltech101 as a labeled source dataset. It is important to point out that we do not use paired sensor data even though it is available for N-Caltech101 and Caltech101. As our approach can leverage unpaired single images for event-based training, we extend the frame-based Caltech101 with a set of additional images showing the 101 classes. These additional images were also used in the baseline method VID2E [6] for training a classification network on events, which are generated based on simulated motion.

Our task network for classification is inspired by Resnet18 [41]. In particular, we use the first layers up to the third residual block of Resnet18 without the first max-pooling layer for both sensor encoders. The second of these residual blocks is shared between the event and image encoder. The task network consists of the remaining Resnet18 layers. All the Resnet18 layers were initialized with weights pre-trained on ImageNet [8]. The architectures proposed in Drit++ [12] were adopted for the sensor-specific encoder, decoder, shared latent discriminator, and event frame discriminator. While several modules are used during training, crucially, during testing, we only use the fast ResNet-18. which needs 4.5ms to process one sample on a Nvidia Quadro RTX 8000. The event histogram [42] is used as event representation to facilitate the image-to-event translation. We augment the pseudo-flow with random translation fields (Section III-C) as N-Caltech101 only contains translational motions. Similar to supervised learning, we split the target data into training, validation, and test data. Thus, no testing sample was seen during training/validation, neither in the image nor event domain.

Results In addition to the state-of-the-art methods VID2E [6] and E2VID [4] applicable to the UDA setting, we also include supervised methods, which have access to the event labels during training. The classification accuracies are reported in Tab. I.

Our approach outperforms the state-the-art method E2VID by 2.7% in terms of accuracy. Moreover, our inference network is a simple Resnet18, which is computationally much more lightweight than E2VID. Compared to VID2E, our approach achieves a 4.1% higher accuracy. There are two possible reasons for the increased performance. First, our method focuses specifically on task transfer and thus exploits the image and event domain to learn task-relevant features. This multi-modal learning helps to extract more informative features, which was confirmed in recent work [43] as well. Second, VID2E generates events based solely on model assumptions, whereas our approach uses the generative event model combined with a data-driven network to approximate target events. Thus, our network can adjust better to the specific target event data, which is influenced by the event camera model and parameter settings. As shown in Fig. 4, our approach generates realistic event frames based on single grayscale images.

Our approach even outperforms the supervised methods HATS [44] and EST [45], as seen in Table I. One advantage of our method compared to HATS and EST is the increased size of the training dataset. As our approach can use single images without corresponding events, we can easily extend the training dataset with additional class samples, as done for VID2E. This also explains the higher classification accuracy of our approach compared to the supervised setting with the same architecture. Since EvDistill [30] was trained on a different training split for Caltech101 in the image domain, it is hard to compare against this approach. Nevertheless, we report the performance of our approach trained on the complete Caltech101 dataset and the performance reported in [30].

We additionally report the performance of a simple cycle translation UDA framework without the embedding split into shared and event-specific features. The significantly lower performance shows that the feature space split is crucial for the task transfer between images and events. In the case of classification on N-Caltech101, the flow prediction does not provide any improvement compared to a model, which directly predicts an event representation. This can be explained due to the simple planar motion distribution present in the event samples of N-Caltech101. For a more complex motion distribution, i.e., in driving car sequences, the flow prediction almost doubles the object detection performance, see Tab. II.

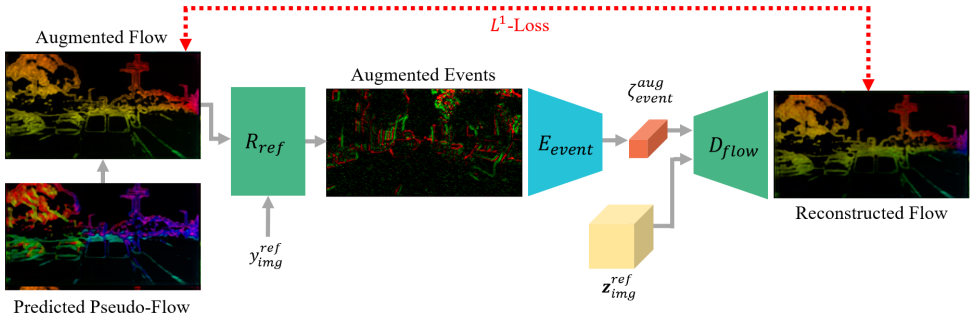

Fig. 3: To enforce the split into sensor shared features and event-specific features, i.e., motion features, we propose to augment the pseudo-flow predictions. Specifically, we take a pseudo-flow prediction and augment the flow with the target domain-specific motions. In the shown car driving case, we sample an augmentation based on random epipoles in the image. Event-specific features $\zeta^{\text{aug}}_{\text{event}}$ are extracted from the events constructed based on the augmented flow. These event specific features $\zeta^{\text{aug}}_{\text{event}}$ are then combined with the shared features $z^{\text{ref}}_{\text{img}}$ from the reference frame $y^{\text{ref}}_{\text{img}}$ to reconstruct the pseudo-flow. An L^1-loss is then applied between the augmented and the reconstructed loss. By doing so, the networks can only adapt the motion features $\zeta^{\text{aug}}_{\text{event}}$ as the content features are fixed.

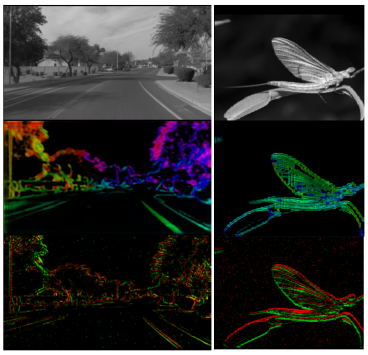

Fig. 4: Images (top) are converted to events (bottom), via intermediate pseudo-flow map (middle) and image gradients.

Method	UDA	Accuracy ↑
E2VID [4]	✔	0.821
VID2E [6]	✔	0.807
Simple Cycle	✔	0.577
Ours w/o Flow	✔	**0.848**
Ours	✔	**0.848**
E2VID [4]	✗	0.866
VID2E [6]	✗	0.906
EST [45]	✗	0.817
HATS [44]	✗	0.642
Ours supervised	✗	0.839
EvDistill* [30]	✔	0.902
Ours*	✔	0.938

TABLE I: Classification accuracies on the N-Caltech101 dataset. The top of the table shows the methods targeting UDA, i.e., they do not have access to the event labels during training. We have also listed methods that use the ground truth labels during training and are thus not applicable for UDA. To stay consistent with the evaluation in [30], we report the performance achieved by our model trained on the whole Caltech101 dataset(*).

Method	Unpaired	mAP ↑
ESIM [11]	✔	0.02
E2VID [4]	✔	0.28
Ours w/o flow	✔	0.26
Ours w/o augm	✔	0.48
Ours w/o split	✔	0.41
Ours	✔	**0.54**
EventGAN [5]	✗	0.30
YOLOv3-GN* [2]	✗	0.70

TABLE II: Mean average precision for the task of object detection on the MVSEC dataset. *Different test labels and trained on the same sequence

B. Object Detection

Experimental Setup In the case of object detection, we evaluate on the Multi-Vehicle Stereo Event Camera Dataset (MVSEC) [15]. The authors of EventGAN [5] provided us with car bounding box labels for the outdoor_day_2 sequence, on which they evaluated EventGAN. For training, we use the two outdoor sequences from MVSEC and add the DDD17 [46] dataset, which contains unlabeled events, both captured with a DAVIS346[47]. As a labeled image dataset, we use Waymo Open Dataset [10]. In addition to MVSEC, we also report detection results on the large-scale, labeled 1 MP automotive detection dataset (1-MP-ADD) [48], which contains labels for 25 million object bounding boxes. Here, we train two methods, one using the labeled training data, and one in the UDA setting, namely, with unlabeled training data and labeled images from Waymo. To stay consistent with other evaluation datasets, we only consider the ”car” class in 1-MP-ADD.

Except for the task branch, we use the same network layers as for the classification task. Similar to EventGAN and the network grafting approach YOLOv3-GN [2], the task branch for object detection consists of YOLOv3 [49] layers. During inference, our network needs 27ms to process one sample on a Nvidia Quadro RTX 8000.

Results We compare against the event simulator ESIM [11] as well as E2VID on the task of object detection. Additionally, YOLOv3-GN and EventGAN are included as paired baselines, i.e., they were trained with events and the corresponding frames. For the performance of YOLOv3-GN, we report the value published in their paper, which was computed on the same outdoor_day_2 sequence, but with different bounding boxes generated by a frame-based object detector applied to the grayscale images. The object detection performances on MVSEC are reported in Table II as mean Average Precision (mAP) [9]. Compared to approaches trained on unpaired data, our approach achieves the highest performance, outperforming the next best method [36] by 26% in terms of mAP. We credit the better performance to the fact that, while E2VID is only concerned with event-to-image translation, our method explicitly optimizes for the task objective, thus generating more task optimized representations. The low performance of the detector trained using ESIM can be explained by the large domain gap between the generated and real events. The events from ESIM were generated with a uniform planar motion [5], which differs greatly from driving sequence motions.

We also compare our method against paired approaches [5], [2], which use the grayscale images and the corresponding events of the outdoor_day_2 sequence during training. Nevertheless, our method outperforms EventGAN by a significant margin (0.24 mAP). Compared to EventGAN, which generates labeled events from two frames, we focus on the specific task transfer from images to events. By splitting the embedding space into shared and event-specific motion features, we can leverage the labeled images to extract task-specific knowledge in the shared space. Therefore, the task network can focus on task-specific features. Moreover, the significant improvement

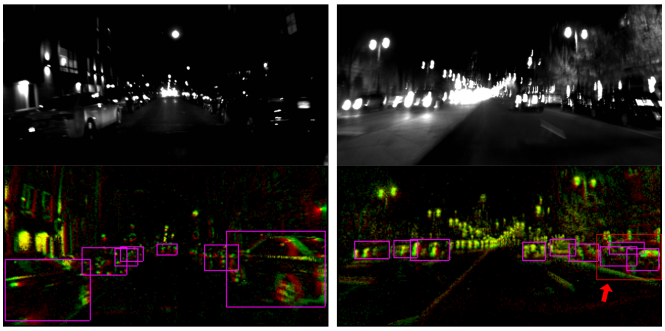

Fig. 5: Our task transfer framework enables to transfer from daylight images to events recorded during the night. The top row shows the VI-sensor images, which are underexposed (left) and suffer from motion blur (right). In contrast, the event histograms (bottom row) include much more details than the images, which helps the predictions of cars (magenta). The bottom right arrow indicates the only three false predictions, which lay on top of just one car.

can be attributed to the combination of our novel flow module (from 0.26 to 0.48), our flow augmentation (from 0.48 to 0.54), and our proposed feature split (from 0.41 to 0.54). As our method combines prior sensor knowledge with adversarial training, we can generate a more realistic event distribution and thus have an advantage at transferring the object detection task from images to events. For YOLOv3GN, a pre-trained Yolov3 network was adapted directly to the frames from the outdoor_day_2 with the corresponding events to align the embedding space with paired data. As the same sequence is used as a test set, a fair comparison is difficult since the reported test score is likely to overestimate the true test score. By contrast, while our method has a 16% lower performance, it is important to note that it does not have access to labels in the target domain and thus solely relies on unpaired images and labels from a vastly different domain. By contrast, the method in [2], achieves limited task transfer since it assumes paired (*i.e.* per-pixel aligned and synchronized) data to work.

Results on 1-MP-ADD The detector trained with our framework achieves 0.26 mAP compared to 0.48 mAP when trained in a supervised fashion on the 1-MP-ADD dataset. Crucially, in this setting, unsupervised domain adaptation is not needed since labeled training data is available (25 million bounding boxes). However, if we evaluate the fully supervised network on MVSEC, it only achieves a performance of 0.49 mAP, which is lower than 0.54 mAP achieved by our UDA approach. This highlights that our approach can more effectively bridge the domain gap between datasets compared to supervised methods.

Daylight Images to Night-Time Events In general, events recorded during the night differ from day-light events [7]. Nevertheless, previous work [48] qualitatively demonstrated that it is possible to transfer a network trained on daylight events to night events without any domain adaptation. A fine-tuning on night events is usually also not possible since there are no labels available for night-time events [48] due to the complicated labeling of night sequences. Different from [48], our UDA framework explicitly leverages unlabeled events recorded during the night and creates a shared embedding space for daylight images and night-time events. Thus, the task network trained with our framework can adapt to the noise distribution specific to nighttime events. To demonstrate this ability, we use the relatively high-quality images from the Waymo Open Dataset to transfer the task of car detection to events recorded during the night, where standard cameras are underexposed. We visualize our detector in this scenario on MVSEC [50] (Fig. 5). Standard frames recorded with a VI-Sensor [51] (top row) are underexposed and blurry, while event data has a higher dynamic range and does not suffer from motion blur. Our method detects all cars present in the event stream (orange boxes), only making a single mistake by misidentifying a cluster of cars (red arrow). Crucially, this method was entirely trained with labeled images in daylight scenarios, without ever seeing a label in the dark. This example highlights the potential of the method for transferring task knowledge to challenging night-time scenarios. The robustness of our approach is also shown in the supplementary video.

C. Future Work

We have shown the effectiveness of our proposed method on three event datasets, including translating and driving motions. To extend our framework to datasets with different flow distributions, multiple potential flow augmentations could be applied such as flipping, rotating, scaling, and constant addition. However, object motion independent of the global scene motion still remains a challenge for our method and is left as future work. Additionally, the current method only uses the event-specific features during image-to-event translation. Their use in the task network remains unexplored, and could be useful for motion-related tasks such as optical flow estimation.

V. Conclusion

Learning for event-based vision has been held back by the scarcity of training data. In contrast, image or video-based methods have tremendously improved in performance due to the availability of large-scale datasets. This work proposes a framework to address this problem by leveraging large-scale image datasets with unsupervised domain adaptation. To achieve this, our method transfers task-specific knowledge from frame-based datasets to the event-based domain without the need for paired sensor data. Therefore, this framework allows models to be trained directly with labeled images and *unlabeled* event data. In conclusion, our work unlocks the potential to use any frame-based dataset to train an event-based network. By large-scale datasets like the extended N-Caltech101 dataset [6] or the Waymo Open Dataset [10], we outperform state-of-the-art method for classification by 2.7% and object detection by 0.26 mAP, in the UDA setting.

References

[1] R. Fisher, "Cvonline: Image databases," https://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm, 2021.
[2] Y. Hu, T. Delbruck, and S.-C. Liu, "Learning to exploit multiple vision modalitiesby using grafted networks," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020.
[3] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
[4] 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.

[5] A. Z. Zhu, Z. Wang, K. Khant, and K. Daniilidis, "Eventgan: Leveraging large scale image datasets for event cameras," *arXiv preprint arXiv:1912.01584*, 2019.

[6] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to Events: Recycling video datasets for event cameras," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.

[7] Y. Hu, S.-C. Liu, and T. Delbruck, "v2e from video frames to realistic dvs events," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2021.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, Apr. 2015.

[9] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, 06 2010.

[10] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] H. Rebecq, D. Gehrig, and D. Scaramuzza, "ESIM: an open event camera simulator," in *Conf. on Robotics Learning (CoRL)*, 2018.

[12] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. K. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation viadisentangled representations," *International Journal of Computer Vision*, 2020.

[13] Z. Zheng, Y. Wu, X. Han, and J. Shi, "Forkgan: Seeing into the rainy night," in *The IEEE European Conference on Computer Vision (ECCV)*, August 2020.

[14] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[15] A. Z. Zhu, D. Thakur, T. Ozaslan, 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, July 2018.

[16] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Front. Neurosci.*, vol. 9, 2015.

[17] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, July 2020. [Online]. Available: https://doi.org/10.1145/3400066

[18] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *in Proc. 32nd AAAI Conf. Artif. Intell., New Orleans, FL, USA, Feb. 2018 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018.

[19] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15, 2015.

[20] P. Li, X. Liang, D. Jia, and E. P. Xing, "Semantic-aware grad-gan forvirtual-to-real urban scene adaption," in *British Mach. Vis. Conf. (BMVC)*, 2018.

[21] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[22] R. Takahashi, A. Hashimoto, M. Sonogashira, and M. Iiyama, "Partially-shared variational auto-encoders for unsupervised domain adaptation with target shift," in *Computer Vision – ECCV 2020*. Springer International Publishing, 2020.

[23] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[24] A. Zanardi, J. Zilly, A. Aumiller, A. Censi, and E. Frazzoli, "Wormhole learning," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[25] J. Yang, N. C. Dvornek, F. Zhang, J. Chapiro, M. Lin, and J. S. Duncan, "Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, 2019.

[26] S. Mostafavi I., L. Wang, Y.-S. Ho, and K.-J. Y. Yoon, "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[27] S. M. Mostafavi I., J. Choi, and K.-J. Yoon, "Learning to super resolve intensity images from events," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020.

[28] 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 *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.

[29] S. Zhang, Y. Zhang, Z. Jiang, D. Zou, J. Ren, and B. Zhou, "Learning to see in the dark with events," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[30] L. Wang, Y. Chae, S.-H. Yoon, T.-K. Kim, and K.-J. Yoon, "Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.

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

[32] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[33] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[34] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle consistent adversarial domain adaptation," in *International Conference on Machine Learning (ICML)*, 2018.

[35] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[36] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Conf. Neural Inf. Process. Syst. (NIPS)*, 2014.

[38] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[39] D. Tran, R. Ranganath, and D. M. Blei, "Hierarchical implicit models and likelihood-free variational inference," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017.

[40] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, 2006.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[42] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[43] N. Sayed, B. Brattoli, and B. Ommer, "Cross and learn: Cross-modal self-supervision," in *German Conference on Pattern Recognition (GCPR) (Oral)*, Stuttgart, Germany, 2018. [Online]. Available: https://arxiv.org/abs/1811.03879v1

[44] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[45] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Int. Conf. Comput. Vis. (ICCV)*, 2019.

[46] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD17: End-to-end DAVIS driving dataset," in *ICML Workshop on Machine Learning for Autonomous Vehicles*, 2017.

[47] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 130dB 3μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, 2014.

[48] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," in *Conf. Neural Inf. Process. Syst. (NIPS)*, 2020.

[49] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv e-prints*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[50] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Robotics: Science and Systems (RSS)*, 2018.

[51] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014.

Supplementary: Bridging the Gap between Events and Frames through Unsupervised Domain Adaptation

Nico Messikommer, Daniel Gehrig, Mathias Gehrig, Davide Scaramuzza

A1 Losses

In the following, we give a detailed explanation of the losses applied in our proposed framework.

A1.1 Adversarial Losses

Latent Space Loss

To ensure that the task branch T can seamlessly transfer between events and frames, we enforce its input, i.e., both latent representation $\mathbf{z}_{\text{img}}$ and $\mathbf{z}_{\text{event}}$, to lay on one manifold. This is done by applying adversarial training [1] with a discriminator network Dis_{lat}. The adversarial training aligns the distribution of $\mathbf{z}_{\text{img}} = E_{\text{img}}(\mathbf{y}_{\text{img}})$ and $\mathbf{z}_{\text{event}} = E_{\text{event}}(\mathbf{y}_{\text{event}})$. Similar to [2], a hinge adversarial loss [3] is adopted.

$$\begin{aligned}\mathcal{L}_{\text{lat.,disc.}} =& \mathbb{E}_{\mathbf{y}_{\text{img}}}[\max(0, 1 - Dis_{\text{lat}}(\mathbf{z}_{\text{img}}))] \\ &+ \mathbb{E}_{\mathbf{y}_{\text{event}}}[\max(0, 1 + Dis_{\text{lat}}(\mathbf{z}_{\text{event}}))] \\ \mathcal{L}_{\text{lat.,gen.}} =& \mathbb{E}_{\mathbf{z}_{\text{img}}}[Dis_{\text{lat}}(\mathbf{z}_{\text{img}})] - \mathbb{E}_{\mathbf{y}_{\text{event}}}[Dis_{\text{lat}}(\mathbf{z}_{\text{event}})].\end{aligned} \tag{1}$$

Image-to-event Translation Loss

As an additional constraint, we force the latent representations $\zeta_{\text{event}}, \mathbf{z}_{\text{img}}$ to carry sufficient information, such that they can be decoded into an artificial event-frame. To do this, we combine the event-specific features ζ_{event} from a random event representation $\mathbf{y}_{\text{event}}$ and the content features $\mathbf{z}_{\text{img}}$ from an image $\mathbf{y}_{\text{img}}$ to generate artificial events $\hat{\mathbf{y}}_{\text{event}} = R_{\text{ref}}(D_{\text{flow}}(\mathbf{z}_{\text{img}}, \zeta_{\text{event}}))$ using a pseudo-flow decoder D_{flow} and a refinement net R_{ref}. These events share the image content but contain the event-specific features, e.g., motion distribution, of the reference events $\mathbf{y}_{\text{event}}$. The following adversarial loss is applied on the image-to-event reconstruction $\hat{\mathbf{y}}_{\text{event}}$. Similar to the embedding space alignment, a PatchGAN [4] discriminator and the hinge loss [3] is adopted for the sensor translation.

$$\begin{aligned}\mathcal{L}_{\text{recons.,disc.}} =& \mathbb{E}_{\hat{\mathbf{y}}_{\text{event}}}[\max(0, 1 - Dis_{\text{event}}(\hat{\mathbf{y}}_{\text{event}}))] \\ &\mathbb{E}_{\mathbf{y}_{\text{event}}}[\max(0, 1 + Dis_{\text{event}}(\mathbf{y}_{\text{event}}))] \\ \mathcal{L}_{\text{recons.,gen.}} =& \mathbb{E}_{\hat{\mathbf{y}}_{\text{event}}}[Dis_{\text{event}}(\hat{\mathbf{y}}_{\text{event}})]\end{aligned} \tag{2}$$

A1.2 Translation Consistency

By translating from single images to events, we can formulate consistency losses on the latent variables $\mathbf{z}_{\text{img}}$ and ζ_{event} as summarized below:

$$\begin{aligned}\mathcal{L}_{\text{cycle}} =& |\mathbf{z}_{\text{img}} - E_{\text{event}}(R_{\text{ref}}(D_{\text{flow}}(\mathbf{z}_{\text{img}}, \zeta_{\text{event}})))|_1 \\ &+ |\zeta_{\text{event}} - E_{\text{event, attr}}(R_{\text{ref}}(D_{\text{flow}}(\mathbf{z}_{\text{img}}, \zeta_{\text{event}})))|_1\end{aligned} \tag{3}$$

A1.3 Flow Augmentation Loss

As shown in Figure 4 in the manuscript, we augment the flow by combining the augmented, event-specific features $\zeta^{\text{aug}}_{\text{event}}$ with the shared features $z^{\text{ref}}_{\text{img}}$ from the reference frame $y^{\text{ref}}_{\text{img}}$. The resulting event representation should be identical to the events generated by the augmented flow. By enforcing this identity, the network can only store the augmented flow information in the event-specific features $\zeta^{\text{aug}}_{\text{event}}$ since the shared features are constant.

$$\begin{aligned}\mathcal{L}_{\text{augm}} =& |E_{\text{event, attr}}(y^{\text{aug}}_{\text{event}}) \\ &- E_{\text{event, attr}}(R_{\text{ref}}(D_{\text{flow}}(z^{\text{ref}}_{\text{img}}, \zeta^{\text{aug}}_{\text{event}})))|_1\end{aligned} \tag{4}$$

A1.4 Flow Smoothness

As common in the literature, we add a smoothness loss $\mathcal{L}_{\text{smooth}}$ to further constrain the pseudo-flow prediction. The smoothness loss consists of the Charbonnier loss function [5] applied to the difference of a flow vector with its eight neighboring flow vectors (including diagonal neighbors):

$$\mathcal{L}_{\text{smooth}} = \sum_{\mathbf{x}} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \rho(\mathbf{v}_{\mathbf{x}} - \mathbf{v}_{\mathbf{y}}) \tag{5}$$

Where $\mathbf{x} = (x, y)$ are image pixels, $\mathcal{N}(\mathbf{x})$ denotes the 8 neighbors of $\mathbf{x}$ and $\rho(x) = (\epsilon^{\alpha} + x^{\alpha})^{\frac{1}{\alpha}}$. We use $\alpha = 0.45$ and $\epsilon = 0.001$.

A1.5 Image Gradient Loss

To prevent mode collapse of the image-to-event generation, a gradient loss $\mathcal{L}_{\text{grad}}$ is applied on the event reconstructions. This loss penalizes areas that have few events and high image gradient as events are generated mainly by image gradients. We enforce this constraint by introducing the following loss:

$$\mathcal{L}_{\text{grad}} = \sum_{(x,y)} \max(0, 0.7 - N_{(x,y)}) |\nabla I_{(x,y)}| \tag{6}$$

Where we sum over all pixels that have an image gradient magnitude $|\nabla I_{(x,y)}| > 0.7$. Here $N_{(x,y)} \in [0, 1]$ represents the normalized number of events per pixel, as predicted by the event generation module. Without this loss, the event generation focuses too strongly on the noisy and low-quality events present in the real event domain. This noise prediction helps the generator fool the discriminator since it can just

The authors are with the Robotics and Perception Group, Department of Informatics, University of Zurich, and Department of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland (http://rpg.ifi.uzh.ch). This work was supported by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation (SNSF).

predict noisy event frames that do not contain any structural information. Due to these noisy event frames, the task transfer performance degenerates. Therefore, we mitigate this effect by introducing the proposed image gradient loss.

A1.6 Task Loss

For the task loss $\mathcal{L}_{\text{task}}$, we use the standard cross-entropy loss and the loss defined in Yolov3 [6] for classification and object detection, respectively. As illustrated with $\mathcal{L}_{\text{task}}$ in Fig.3 in the manuscript, the task loss is applied twice, once in the image domain and once in the translated event domain.

A1.7 Training Procedure

The final loss is presented in Equation 7. As common in adversarial training, we train the generator and discriminator in separate steps. The network parameters are updated alternatingly by minimizing the following losses, whereby we train the generator for one step after two discriminator training steps.

$$\begin{aligned} L_{Gen} =& \mathcal{L}_{\text{lat,gen.}} + \mathcal{L}_{\text{recons.,gen.}} + \mathcal{L}_{\text{cycle}} \\ &+ 2\mathcal{L}_{\text{augm}} + \mathcal{L}_{grad} + \mathcal{L}_{task} \\ &+ \mathcal{L}_{\text{smooth}} \\ L_{Dis} =& \mathcal{L}_{\text{lat,disc.}} + \mathcal{L}_{\text{recons.,disc.}} \end{aligned} \tag{7}$$

A2 Ablation

We ablated our proposed design choices with multiple experiments for the task of classification on N-Caltech101 [7] as well as for the task of object detection on MVSEC [8].

A2.1 Classification

The experiments conducted on N-Caltech101 underline the improved performance of our proposed transfer framework compared to standard *Unsupervised Domain Adaptation (UDA)* methods, as reported in Table I.

In a first experiment, we have adapted our network to a simple cycle GAN framework [9], in which we predict events and grayscale images directly from a shared feature space (Simple cycle). The significantly lower performance of 0.577 compared to 0.848 of our final approach verifies the need of considering prior sensor knowledge and the feature space split into a shared embedding space and an event-specific space for motion information.

In a second experiment, we use our final network to solely translate from grayscale images to events (Ours transl). This way, we generate a labeled event dataset on which a task network can be trained. The performance of 0.832 verifies the accurate single image-to-event translation of our final network. However, the lower performance compared to our final task transfer network confirms that the task network benefits from simultaneously learning on images and events to detect the most relevant task features. A similar conclusion was also drawn in [10], where the authors achieved an increased task performance by learning with paired images and optical flow frames.

Method	UDA	Accuracy ↑
Simple cycle	✔	0.577
Ours transl	✔	0.832
Ours w/o ref	✔	0.592
Ours w/o flow	✔	**0.848**
Ours	✔	**0.848**
Ours supervised	✗	0.839

TABLE I: Ablation for the classification task on the N-Caltech101 dataset.

Method	Unpaired	mAP ↑
Ours w/o flow	✔	0.26
Ours w/o split	✔	0.41
Ours w/o augm	✔	0.48
Ours	✔	**0.54**

TABLE II: Ablation for the object detecion task on the MVSEC dataset.

For the validation of our refinement network, we report the classification accuracy achieved without adding the residual event representation to the events constructed based on the flow and image gradients (Ours w/o ref). As expected, the performance suffers a substantial drop compared to the final network, which shows the importance of the refinement network. Due to the simple planar motion distribution, the event generation based on pseudo flow achieves the same performance as a direct prediction of the event representation (Ours w/o flow).

The benefits of the event generation based on the generative event model for more complex motions are shown in the experiments conducted on MVSEC. Finally, as reported in the manuscript in Section M4.1, our transferred network achieves higher performance than the same architecture trained with ground truth labels. This can be explained mainly for two reasons. First, our UDA method allows us to include additional labeled image data to help train the event classification network. Second, the simultaneous learning in the event and image representation increases the overall task performance.

A2.2 Object Detection

The ablation experiments for the task of objection detection on MVSEC strongly confirm our proposed network modules. The first experiment (Ours w/o flow) shows that the image-to-event translation without the generative event model decreases the task transfer by a large margin. The direct event representation prediction without the pseudo-flow estimation is not able to capture the complex motion distribution of a driving car sequence. The second experiment (Ours w/o split) verifies the benefits of splitting the feature space into sensor shared and event-specific features. Without the embedding space split, the performance suffers a significant mAP drop from 0.54 to 0.41 In the third experiment (Ours w/o augm), we validate the introduced flow augmentations, which help to split the embedding space into shared and event-specific features. The flow augmentations improve the detection score by 0.06 mAP. In conclusion, each of our proposed network modules substantially improves the object detection performance.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Conf. Neural Inf. Process. Syst. (NIPS)*, 2014.

[2] A. Z. Zhu, Z. Wang, K. Khant, and K. Daniilidis, "Eventgan: Leveraging large scale image datasets for event cameras," *arXiv preprint arXiv:1912.01584*, 2019.

[3] D. Tran, R. Ranganath, and D. M. Blei, "Hierarchical implicit models and likelihood-free variational inference," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[5] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int. J. Comput. Vis.*, 2014.

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv e-prints*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[7] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Front. Neurosci.*, vol. 9, 2015.

[8] A. Z. Zhu, D. Thakur, T. Ozaslan, 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, July 2018.

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[10] N. Sayed, B. Brattoli, and B. Ommer, "Cross and learn: Cross-modal self-supervision," in *German Conference on Pattern Recognition (GCPR) (Oral)*, Stuttgart, Germany, 2018. [Online]. Available: https://arxiv.org/abs/1811.03879v1