Unsupervised Domain Adaptation for Training Event-Based Networks Using Contrastive Learning and Uncorrelated Conditioning

Dayuan Jian
University of Southern California
djian@usc.edu

Mohammad Rostami
University of Southern California
rostamim@usc.edu

Abstract

Event-based cameras offer reliable measurements for preforming computer vision tasks in high-dynamic range environments and during fast motion maneuvers. However, adopting deep learning in event-based vision faces the challenge of annotated data scarcity due to recency of event cameras. Transferring the knowledge that can be obtained from conventional camera annotated data offers a practical solution to this challenge. We develop an unsupervised domain adaptation algorithm for training a deep network for event-based data image classification using contrastive learning and uncorrelated conditioning of data. Our solution outperforms the existing algorithms for this purpose.

1. Introduction

Inspired by biological visual systems, event-based cameras are designed to measure image pixels independently and asynchronously through recording changes in brightness, compared to a reference brightness level [25]. A stream of events is measured which encodes the location, time, and polarity of the intensity changes. Compared to conventional frame-based cameras, they offer high dynamic range, high temporal resolution, and low latency which make them ideal imaging devices for preforming computer vision (CV) tasks that involve high dynamic range and fast movements. Despite this promising prospect, adopting artificial intelligence and particularly deep learning to automate CV tasks in this domain is challenging. The primary reason is that training deep neural networks relies on manually annotated large datasets [49]. However, annotated event-data is scarce due to the recency of these cameras: event-based data represent only 3.14% of existing vision data [18].

Unsupervised domain adaptation (UDA) [10] is a learning framework to train a model for a target domain with unannotated data through transferring knowledge from a secondary source domain with annotated data. The core idea in UDA is to reduce the distributional domain gap by mapping data from both domains into a shared domain-agnostic embedding space [1, 2, 55, 36, 45, 46, 47]. Although event-based and frame-based cameras are significantly different, the intrinsic relationship between their outputs through the real-world relates the information content of the outputs considerably. As a result, a frame-based annotated dataset can be served as a source domain for an event-based domain as the target domain in the context of a UDA problem. Even if the related frame-based data is unannotated, annotating frame-based data is much simpler.

The above possibility has led to a line of UDA algorithms for event-based CV tasks. Initial works in this area consider datasets with paired events and frames, i.e., the pixel-level recordings describe the same input [18, 63]. Despite being effective, these methods are highly limited because they consider that both measurement types are preformed simultaneously. Recently, a few UDA algorithms have been developed for unpaired event-based data [43, 32]. An effective idea is to use the idea of translation in UDA [34] and generate events for corresponding video data [12, 19]. An alternative idea is to generate event-based annotated data synthetically [41] and use it as the source domain [39].

We develop a new UDA algorithm for event-based data using unpaired frame-based data. Our contribution is twofold. First, we use contrastive learning in combination with data augmentation to improve the model generalizability. The idea is to project different augmentation types of one object to a latent representation and train an encoder to maintain identities for all augmentation types. Second, we introduce an uncorrelated conditioning loss term. The new loss term is used to regularize the model with additional information: the latent vector representation of an object under event cameras should be uncorrelated to how the object looks like under event cameras. Experiments show that our method leads to learning a better representation of events in an unsupervised fashion. Our results indicate 2.0% on Caltech101 → N-Caltech101 and 3.7% on CIFA10 → CIFA10-DVS performance improvements over state-of-the-art.

2. Related Work

Unsupervised Domain Adaptation for Frame-Based Tasks: UDA has been explored extensively in CV when both domains are frame-based measurements from the same modality [59]. The core idea for UDA in frame-based tasks is to train an encoder that is shared across the two domains and map the training data points from both domains into the shared domain-agnostic latent embedding space, i.e., modeled as the encoder output space. As a result, if we use the source domain labeled data and train a classifier that receives its input from the embedding space, it will generalize well on the target domain. The domain-agnostic embedding has been learned primarily using either generative adversarial learning [17, 50, 37, 62, 28, 30] or by minimizing a suitable distribution alignment loss term [27, 11, 29, 21, 48, 53, 44] in the UDA literature. In the first approach for UDA, the generator subnetwork within the generative adversarial learning architecture [14] is served as the shared encoder which is trained to compete against a discriminative subnetwork to learn a domain-agnostic, yet discriminating embedding space. As a result of this process, the two distributions are aligned indirectly at the output of the generative subnetwork. In the second approach, the distance between the empirical distributions of the two domains is measured at the output of the shared encoder, e.g., using sliced Wasserstein distance distance [52, 54, 54], and the encoder is then trained such that that distance is minimized directly.

Unsupervised Domain Adaptation for Event-Based Tasks: despite the challenge of annotated data scarcity for event-based data, only recently some works have tried to benefit from UDA framework to address tasks with unannotated event-based data. The reason is that when the source and the target domain have different modalities, e.g., frame-based and event-based data, the distribution gap between the two domains is significantly larger. For this reason, using a shared encoder may not be applicable for aligning the distributions. However, we can benefit from the special case of image-to-image translation [34] from the UDA literature. The core idea of image-to-image translation is to map images from either of the two domains to the other domain using a generative model. As a result, we can match the distributions directly at the input image space of one of the domains. For example, if we translate the target domain images to the source domain, we can directly use the source-trained classifier. For this reason, a group of exiting works translate event data frame-based images using adversarial training [57, 58, 7]. A major limitation for these works is that we require paired image-event data that is measured on the same scene to learn cross-domain correspondences. This is a strong limitation because it means

that we should measure and generate paired data to build new datasets which in some applications is not even feasible, e.g., when event and frame sensors are used in different lighting conditions. This requirement has been relaxed by separating domain-agnostic features, e.g. geometric structures, from the domain-specific feature, e.g., texture [61]. Another existing approach to tackle this challenge is to benefit from the idea of video-to-event translation using generative models with proper architecture. The goal is to convert frame-based video into synthetic event-based data to train the model directly in the event space. To this end, we can either rely on model-based translation [43, 12, 19] or data-driven translation [63]. These methods are still limited because they can only translate videos to events, thus making them inapplicable to image datasets which form the majority of existing datasets. The work most similar to our work splits the embedding space into shared and sensor-specific features. An event generation model is then used to align both domains [32]. In our work, we benefit from contrastive learning to improve model generalizability and define a novel loss term that makes representations of objects uncorrelated to their even-based measurements. Our experiments demonstrate that our contributions can help to achieve the performance level of supervised learning using UDA, relaxing the need for data annotation in some cases.

3. Domain Adaptation from Frames to Events

Our goal is to train a model on labeled frames (RGB images) and then use the trained network on events data, without requiring any labeled events. More importantly, we assume that paired frames and event data is not accessible for our task of interest. Figure 1 presents samples of paired frame and event data for two common event-based datasets. As it can be seen, the pairs are significantly more different than benchmarks that are used in the frame-based UDA literature, e.g., Office-Home [56] or DomainNet [38] dataset. For this reason, UDA is a challenging problem for tasks that have event-based and frame-based data as their target and source domains. Note, however, event and frame-based cameras capture scene in physical work that share strong information overlap which makes knowledge transfer across these two domain feasible. The idea that we will explore is to align the content representations obtained from events to the ones from frames so that we can use the frame classifier.

More specifically, consider that we have access to unannotated event data instance $\boldsymbol{y}_e$ along with annotated frame-based image data $\boldsymbol{y}_f$ that are drawn from the event domain $\mathcal{Y}_e$ and the frame domain $\mathcal{Y}_f$, respectively. Our goal is to map $\boldsymbol{y}_e$ and $\boldsymbol{y}_f$ into a shared domain-agnostic embedding space $\mathcal{Z}$. We refer to this space as content space because the goal is to learn it such that shared information content can be represented in the shared embedding space. Let $E_f(\cdot), E_{e,cont}(\cdot)$ be two suitable deep encoders that project
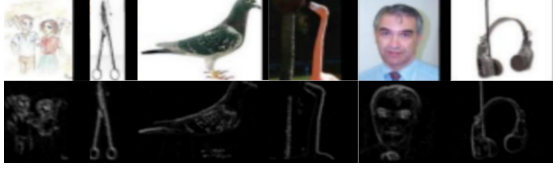
Figure 1: Randomly drawn frame (row 1, Caltech101[60]) and event (row 2, N-Caltech101[35]) pairs

frames to the content feature space and events to the content feature space. We use two independent encoders due to the large domain gap between these two domains. Also, let $C(\cdot)$ denote a classifier network that maps the extracted content features to the label space. Since frames and their labels are provided, we can easily use standard supervised learning to train the neural network $C(E_f(\boldsymbol{y}_f))$ using empirical risk minimization for predicting frame labels.

Our ultimate goal is to train a model for event domain. However, the above approach is not applicable to the event domain because the labels are missing. The idea that we would like to use is to align event content representation $E_{e,cont}(\boldsymbol{y}_e)$ with frame content representation $E_f(\boldsymbol{y}_f)$. If we have access to paired event-frame data, we can just simply use a point-wise loss function to align the data points, but we are considering a more practical setting, where we don't have frame-event pairs. We develop an algorithm for this purpose. Upon aligning the two domains, we can classify the even data using the model $C(E_{e,cont}(\boldsymbol{y}_e))$.

4. Proposed Method

The architecture of our framework is presented in Figure 2. Our core idea is to benefit from adversarial training and generate fake events from content features of frame images such that they look like real events. As a byproduct of generative adversarial learning, the content features from both domain are aligned indirectly. As shown in Figure 2, an image encoder and an event encoder generate features that represent the content inside images and events, respectively. The output space of these two encoders share the same dimension. The content features are then fed into the classifier to predict labels. The goal is to train these encoders such that event content representations have the same distribution of the frame content representations in $\mathcal{Z}$. Additionally, we have a second event encoder $E_{e,att}(\cdot)$ that generates feature attributes that intuitively correspond to how objects look like under an event camera. The frame content and the event attribute are sent to a decoder to generate a fake event. The fake event and the frame are sent to a refinement network to get a fake event with a higher quality. This fake event should contain the content inside the frame and present the content under an event camera. Clearly, if the fake event looks realistic, it means that the shared content

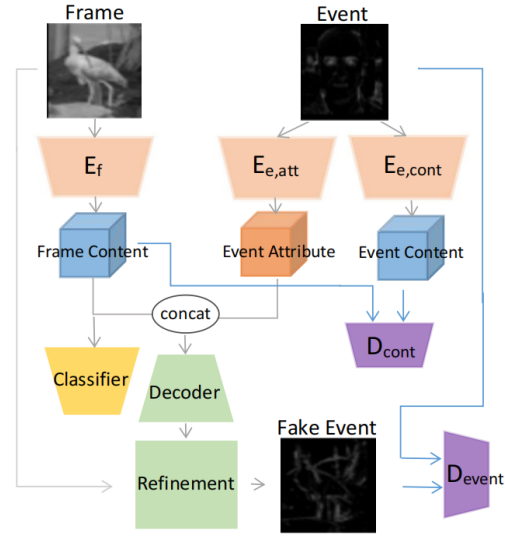encodes shared information from events and frames.

Figure 2: Architecture for our method during training. During testing, only event content encoder and classifier are needed.

To close the adversarial learning architecture, we also use a content discriminator network $D_{cont}(\cdot)$ to discriminate whether the content representation is from an event or an image. This network is trained against both encoders to learn domain-agnostic content features. To enforce more constraints, we also rely on an idea similar to cycleGAN [64]. The generated fake event is fed into the two event encoders and its event attribute and event content are computed. They are compared with the ones which generated the fake event using L_1 loss. To generate fake events that are realistic, we consider a second discriminator network $D_{event}(\cdot)$ used to discriminate real and fake events. Since the fake event is generated from content features which we know its label, the content feature of the fake event is also sent to the classifier for further training. The choice for encoders and the classifier is a design choice. We can also use similar architectures for the three encoders for simplicity. To implement the above constraints, we define the following loss terms and solve the underlying cumulative optimization problem according to the architecture presented in Figure 2:

- **Classification loss on frames,** $\mathcal{L}_{cls,frame}$, assumed to be a Cross Entropy loss function.

- **Classification loss on fake events,** $\mathcal{L}_{cls,fake}$, Cross Entropy loss on classification of fake events.

- **Decoder output loss,** $\mathcal{L}_{decoder}$: it consists of two terms. The first term approximates the event version of a frame image according ℓ_1 loss by the work of Gallego, Guillermo et al[9]. The second term enforces cycle consistency [64] for improvement. The decoded

outputs of event attribute of fake events and frame contents are enforced to be close to the decoded output of event attributed of real event and frame content.

- **Cycle loss on feature representations of contents and attributes:** $\mathcal{L}_{cyc,cont}$ and $\mathcal{L}_{cyc,att}$ These are ℓ_1 losses imposed on the features used to generate the fake events and the features extracted from them:

$$\mathcal{L}_{cyc,cont} = ||E_f(\boldsymbol{y}_f) - E_{e,cont}(fake)||_1$$
$$\mathcal{L}_{cyc,att} = ||E_{e,att}(\boldsymbol{y}_e) - E_{e,att}(fake)||_1 \quad (1)$$

- **Content Discriminator:** There is a discriminator network to differentiate frame content and event content. The purpose of this discriminator is to make frame content features and event content features have the same distribution. Denoted $g(\cdot) = \log(sigmoid(\cdot))$ and $h(\cdot) = \log(1 - sigmoid(\cdot))$. We use relativistic average discriminator[20] loss as follows:

$$\mathcal{L}_{dis,cont} = \frac{1}{N}\sum$$
$$\{\boldsymbol{g}[D_{cont}(E_f(\boldsymbol{y}_f)) - \frac{1}{N}\sum D_{cont}(E_{e,cont}(\boldsymbol{y}_e))] + \quad (2)$$
$$\boldsymbol{h}[D_{cont}(E_{e,cont}(\boldsymbol{y}_e)) - \frac{1}{N}\sum D_{cont}(E_f(\boldsymbol{y}_f))]\}$$

- Generative Loss for encoder $E_f, E_{e,cont}$ as follows:

$$\mathcal{L}_{encoder,cont} = \frac{1}{N}\sum$$
$$\{\boldsymbol{h}[D_{cont}(E_f(\boldsymbol{y}_f)) - \frac{1}{N}\sum D_{cont}(E_{e,cont}(\boldsymbol{y}_e))] + \quad (3)$$
$$\boldsymbol{g}[D_{cont}(E_{e,cont}(\boldsymbol{y}_e)) - \frac{1}{N}\sum D_{cont}(E_f(\boldsymbol{y}_f))]\}$$

- **Event Discriminator loss:** There is a discriminator network to differentiate fake events and real events. The purpose of this discriminator network is to make fake events and real events have the same distribution. We use a relativistic average discriminator[20]:

$$\mathcal{L}_{dis,e} = \frac{1}{N}\sum\{\boldsymbol{g}[D_e(\boldsymbol{y}_e) - \frac{1}{N}\sum D_e(fake)]$$
$$+ \boldsymbol{h}[D_e(fake) - \frac{1}{N}\sum D_e(\boldsymbol{y}_e)]\} \quad (4)$$

- Similarly, the generator loss is:

$$\mathcal{L}_{gen,e} = \frac{1}{N}\sum\{\boldsymbol{h}[D_e(\boldsymbol{y}_e) - \frac{1}{N}\sum D_e(fake)] +$$
$$\boldsymbol{g}[D_e(fake) - \frac{1}{N}\sum D_e(\boldsymbol{y}_e)]\} \quad (5)$$

- **Orthogonal Normalization:** $\mathcal{L}_{orth}$, introduced in BigGAN [3], and is imposed on discriminators. This regularization term is crucial to stabilize the training.

$$L_{orth} = \beta||W^T W \odot (\mathbb{1} - I)||_F^2, \quad (6)$$

where W is the weight matrix, $\mathbb{1}$ is the matrix where all elements are 1, I is the identity matrix, $\odot$ is element-wise multiplication, and denotes $||\cdot||_F$ Frobenius norm.

We can minimize sum of all or a reasonable subset of above loss terms for domain alignment [39]. In this work, we propose two novel ideas based on contrastive learning and uncorrelated conditioning for further improvement.

- **Constrastive Learning:** Figure 3 visualizes our pipeline for computing this term.
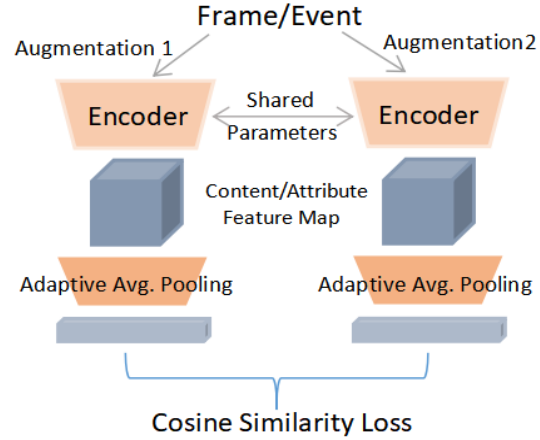

Figure 3: Contrastive learning loss, used for all 3 encoders.

The goal is to use augmentation and generate different versions of an image and enforce them to share the same feature representations in the latent space [15, 5, 6]. Contrastive learning loss serves as a constraint to help convergence to a better solution. While constrastive learning algorithms usually use negative samples as well, we only consider positive samples. Denote $f(\cdot)$ as encoder plus average pooling. Then $f(\boldsymbol{y})$ and $f(\boldsymbol{y}')$ will be two feature vector representations of a single input $\boldsymbol{y}$, produced using augmentation. We then enforce the following hard equality constraint:

$$\frac{\langle f(\boldsymbol{y}), f(\boldsymbol{y}')\rangle}{||f(\boldsymbol{y})||_2 \cdot ||f(\boldsymbol{y}')||_2} = 1, \quad (7)$$

where $\langle \cdot, \cdot \rangle$ denotes vector inner product and $|| \cdot ||_2$ is l_2-norm. We enforce this constraint on outputs of all the three encoders for data from both domains.

- **Uncorrelated Conditioning:** to the best of our knowledge, imposing this constraint on two different features which extracted from the same input is unprecedented. In most tasks, we mostly care about objects captured in an image. The intuition behind this loss is to benefit the full information content of an image. In addition to objects in an image, e.g., a pedestrian, we can deduce
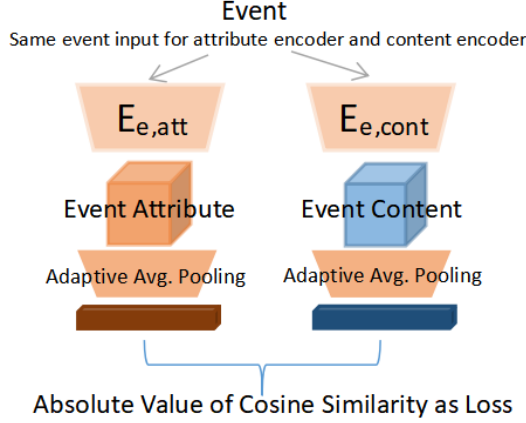
Figure 4: Uncorrelated Conditioning

$$\min \sum_{\boldsymbol{y}_e, \boldsymbol{y}_f} -\frac{\lambda_1 \cdot \langle f_{e,a}(\boldsymbol{y}_e), f_{e,a}(\boldsymbol{y}'_e)\rangle}{||f_{e,a}(\boldsymbol{y}_e)||_2 \cdot ||f_{e,a}(\boldsymbol{y}'_e)||_2}$$
$$-\frac{\lambda_2 \cdot \langle f_{e,c}(\boldsymbol{y}_e), f_{e,c}(\boldsymbol{y}'_e)\rangle}{||f_{e,c}(\boldsymbol{y}_e)||_2 \cdot ||f_{e,c}(\boldsymbol{y}'_e)||_2} - \frac{\lambda_3 \cdot \langle f_{fr}(\boldsymbol{y}_f), f_{fr}(\boldsymbol{y}'_f)\rangle}{||f_{fr}(\boldsymbol{y}_f)||_2 \cdot ||f_{fr}(\boldsymbol{y}'_f)||_2}$$
$$+ \lambda_4 \cdot |\frac{\langle f_{e,a}(\boldsymbol{y}_e), f_{e,c}(\boldsymbol{y}_e)\rangle}{||f_{e,a}(\boldsymbol{y}_e)||_2 \cdot ||f_{e,c}(\boldsymbol{y}_e)||_2}|, \tag{9}$$

where λ_1, λ_2, λ_3 and λ_4 are regularization parameters.

5. Experimental Results

Our implementation code is included as a supplement.

5.1. Experimental Setup

Datasets: We evaluated our approach on N-Caltech101[35] and CIFAR10-DVS[23] event dataset.

- N-Caltech101[35] is an event classification dataset which is the event-based version of the Caltech101[60] dataset. Hence, we can define a UDA task to adapt a model from Caltech101 to N-Caltech101. The N-Caltech101 dataset events are paired with the Caltech101 dataset RGB images. During training, the input RGB images and events are not paired and the label of events are not provided either. Both Caltech101 and N-Caltech101 are split into 45% training set, 30% validation set, and 25% test set. Our test set set is exactly the same as previous works. We reduced the size of training set due to our very tight computing resources.

- CIFAR10-DVS[23] dataset is the event version of CIFAR10[22]. There exists a huge gap between CIFAR10 and CIFAR10-DVS, i.e. CIFAR10 is a relatively easy dataset while CIFAR10-DVS is much more challenging. The large gap poses difficulty to perform UDA. Additionally, CIFAR10 and CIFAR10-DVS datasets are not paired. To our best knowledge, we are the first one to try UDA approach on CIFAR10-DVS. Following the literature, both CIFAR10 and CIFAR10-DVS are split into 5/1 training/testing splits.

Data Augmentation for Contrastive Learning: To compute the constrastive learning loss, we randomly applied color jitter, Gaussian blur, random resize, random affine, random crop, and random flip on frame images to generate the augmented versions. In the events domain, we used random event shift, random flip, random resize, and random crop. These diverse transformations help improve the generalizability of our model and increasing model robustness with respect to domain shift in the input spaces.

other information from the totality of an image. For example, whether the image is blurred or not, whether the image is taken at a day or a night, whether the camera is high-resolution or low-resolution, whether it is taken by a frame camera or an event camera, etc. These are also features we can extract from images. However, these properties are uncorrelated to each other, i.e., whether the image is blurred or not can tell nothing about whether there is a pedestrian in the image, whether the image is taken at a day or a night can tell nothing about the quality of the used. Our idea is to train the model not only to extract features about objects inside an image, but also to extract domain features. Hence, we need to extract features from event data because objects captures by event cameras look different from the same objects captured by a frame cameras. This is the major reason that we produce fake or translated event images. Therefore, we need to impose uncorrelated constraint on the object features and event camera features, e.g., whether an object is a pedestrian or something else in the frame image can tell nothing about how the corresponding event image would look like. Similarly, how an event image looks like would reveal no information about the content of the event image. We enforce this intuition using the uncorrelated conditioning constraint:

$$\frac{\langle f_{e,a}(\boldsymbol{y}_e), f_{e,c}(\boldsymbol{y}_e)\rangle}{||f_{e,a}(\boldsymbol{y}_e)||_2 \cdot ||f_{e,c}(\boldsymbol{y}_e)||_2} = 0, \tag{8}$$

where $f_{e,a}(\boldsymbol{y}_e)$ and $f_{e,c}(\boldsymbol{y}_e)$ are the two feature vector representations of attribute and content, respectively, for the same event $\boldsymbol{y}_e$. The constraint in Eq. 8 helps to decorrelate attribute and content for events.

We further denote $f_{fr}(\boldsymbol{y}_f)$ as the content vector for frame $\boldsymbol{y}_f$. We add the above two terms to the optimization objective function as two additional regularizing loss terms:

Architecture of Networks: we used the first half of a ResNet18[16], except tath the first Conv layer is changed to allow for different input sizes and max pooling layers are removed. The classifier is assumed to be the second half of ResNet18. Hence, we align the distributions in the middle layer of ResNet. Discriminator networks are simple CNNs.

Optimization: R-Adam optimizer [26] is used with $\beta_1 = 0$, $\beta_2 = 0.999$ with $lr = 10^{-4}$, and an exponential decay rate of 0.95. We train on GeForce RTX 2080 Ti GPU. For Caltech101 and N-Caltech101, the batch size is 7 and for CIFAR10 and CIFAR10-DVS, the batch size is 21.

5.2. Performance Results

We first report our performance on the two benchmarks and compare our results against existing SOTA event-based UDA methods. In our results, We have also included results for a second group of supervised learning algorithms that benefit from transfer learning, but assume event labels are accessible during training. All supervised methods on CIFAR10-DVS designed new models specific to events and used different way to represent events. We have also included a "Baseline" version of our approach which is trained using all loss functions, except contrastive learning and uncorrelated conditioning. We refer to our method as Domain Adaptation for event data using Contrastive learning and uncorrelated Conditioning (DAEC^2) in our tables.

Tables 1 and 2 present our results. As it can be seen from the tables, even a Baseline version of our approach achieves 3.8% and a 3.1% increased performances on N-Caltech101 and CIFAR10-DVS, respectively, over state-of-the-art UDA algorithm. The two novel ideas that we proposed, i.e., contrastive learning loss and uncorrelated condition, lead an additional 2.0% and 3.7% performance boosts, respectively, which puts in a competitive range with supervised learning methods. Specifically, we obtain a performance better than the best supervised model TDBN Resnet19[24] on the CIFAR10-DVS benchmark, despite the fact that TDBN is designed specifically to work well on event data. This is a significant observation for practical purposes because it demonstrates that UDA can be used to train models to preform event tasks without data annotation. Please also note that E2VID and VID2E methods produce videos and use videos as a source domain, whereas we use images that makes our approach computationally less expensive.

5.3. Ablative Experiments

We perform an ablative experiment to demonstrate that our major ideas are crucial for improved performance. Table 3 presents our ablative study experiments using N-Caltech101 dataset. We have included the four possible scenarios for using contrastive learning and uncorrelated conditioning for model adaptation. The baseline denotes using all loss functions, except the two corresponding loss terms.

Method	UDA	ResNet	Test Acc
E2VID[42]	✓	34	82.1
VID2E[33]	✓	34	80.7
BG[32]	✓	18	84.8
Baseline	✓	18	88.6
DAEC^2	✓	18	**90.6**
E2VID[42]	✗	34	86.6
VID2E[33]	✗	34	**90.6**
EST[13]	✗	34	81.7
HATS[51]	✗	34	64.2
TDBN[24]	✗	19	78.6

Table 1: Classification accuracies on the N-Caltech101 dataset. Bold font denotes the best method in each category.

Method	UDA	ResNet	Test Acc
BG(Our Repro.)	✓	18	76.3
Baseline	✓	18	79.4
DAEC^2	✓	18	**83.1**
DART[40]	✗	N/A	65.8
Spike-based BP[8]	✗	N/A	74.8
TDBN[24]	✗	19	**78.0**

Table 2: Classification accuracies on the CIFA10-DVS dataset. Bold font denotes the best method in each category.

Condition	Validation Acc.
Baseline Without Event Attribute Encoder	80.7
Baseline	82.5
With Contrastive Learning Loss	83.6
With Uncorrelated Condition	83.7
With Both	84.4

Table 3: Ablative experiments on N-Caltech101 dataset.

We observe that extracting how objects look like under event cameras is necessary. Each of the two novel ideas improves model performance. When we use both ideas, the performance improves compared to using the two ideas individually. These experiments demonstrate that both of our ideas are important for obtaining an optimal performance.

5.4. Analytical Experiments

Optimal Design: we used performance on the validation set to find the optimal design for our implementation. We explored four possibilities to implement the contrastive learning and uncorrelated conditioning constraints:

- Directly impose constraints on feature maps: We can use ℓ_1 loss to align the two feature maps to enforce

contrastive learning. However, it is not easy to implement uncorrelated conditioning on two feature maps. We just naively used negative ℓ_1 loss.

- Global average pooling on the feature maps: After global average pooling, we get a vector representation. For contrastive learning loss, we tried both ℓ_2 distance and cosine similarity, i.e., either minimize ℓ_2 distance or maximize the cosine similarity (in 4, it is $-\cos$ because the loss is the negative of cosine similarity). For uncorrelated conditioning, the natural choice is cosine similarity. When its value is zero, it means the two vectors are uncorrelated. This means that the loss for this constraint is the absolute value of cosine similarity.

- Adding MLP after global pooling then computing distances: SimCLR[4] showed MLP can help. The constraints are implemented similarly to the previous case.

- Inspired by BYOL [15], we also explored using momentum encoders and momentum MLP. As visualized in Figure 5, the idea is when we have two encoders, only one of them gets updated when gradient back propagates and the other one simply copies the parameters using an exponential moving average. Momentum MLP is used when we have one encoder but two MLPs. One MLP gets updated with back propagation and the other one simply copies the parameters using exponential moving average. BYOL[15] also shows that negative samples are not necessary. Eliminating negative sampling allowed us to implement this condition with our limited computing resources.
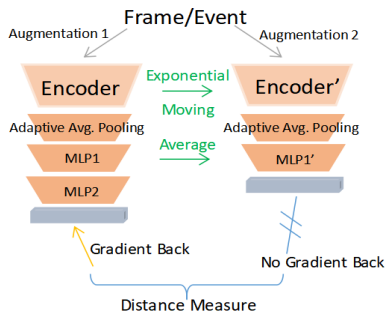

Figure 5: Momentum Encoder, labelled in M_E in table 4, if under average pooling, then there is no MLP1 and MLP1'.

Results for optimal design search are presented in Table 4, where M_E means momentum encoders. If M_E is listed under Avg. Pl., then there is no MLP1 and MLP1' in Figure 5. M_M stands for momentum MLP layers. For all MLPs, if not specified, the same MLP is used for event content and frame content, and a different MLP is used on event attribute. S means using the same MLP for event content, frame content, and event attribute. From the table, we

Contrastive Learning Loss Alone					
	No Proj.	Avg. Pl.		MLP	
Loss	ℓ_1	$-\cos$	ℓ_2	$-\cos$	ℓ_2
Val. Acc	83.5	**83.6**	82.4	83.1	82.7

Uncorrelated Condition Alone			
	No Proj.	Avg. Pl.	MLP
Loss	$-\ell_1$	$\lvert\cos\rvert$	$\lvert\cos\rvert$
Val. Acc	82.2	**83.7**	82.6

Val. Acc		Contrastive	Uncorrelated	
83.5	No Proj.	ℓ_1	$-\ell_1$	Imp.
84.4		$-\cos$		
83.6	Avg. Pl.	ℓ_2	$\lvert\cos\rvert$	
82.7		$-\cos$		M_E
82.4		ℓ_2		M_E
82.2		$-\cos$		
84.0		ℓ_2		
82.9		$-\cos$		M_E
83.3	MLP	ℓ_2	$\lvert\cos\rvert$	M_E
84.2		$-\cos$		M_M
82.7		ℓ_2		M_M
82.8		$-\cos$		S
83.5		$-\cos$		M_MS

Table 4: Optimal design search by validation accuracy on N-Caltech101 dataset for various implementations.

observe that both contrastive learning loss and uncorrelated conditioning loss have positive contributions. However, we conclude that the optimal design for best validation performance is to use average pooling and apply cosine similarity measure to both constraints which leads to 1.1% and 1.2% performance increases, respectively when used separately. Combining them leads to a 1.9% performance increase. However, we see additional MLP layers, momentum encoders and momentum MLP do not lead to further improvement. We think the reason is that an additional MLP gives the model more flexibility to get around with these two constraints. The purpose for these two constraints is to guide encoders extract more informative features, but additional MLP weakens the guidance. Momentum encoders do not help for a similar reason: momentum encoders cannot provide a direct guidance to the encoders. In the case of BYOL[15], momentum encoders are helpful because cosine similarity loss is its objective function, while we only need additional constraints as an extra help for the encoders. We also used a grid search on the weights of the two losses to tune them. We found out the best values are one for both.

2D visualization of features: In order to provide an intuitive justification, we used the UMap [31] visualization tool to reduce the dimension of the data representations in the latent embedding space to two. Figure 6 shows the

Figure 6: UMap for data representations at the model output layers using the test data of CIFAR10 and CIFAR10-DVS.
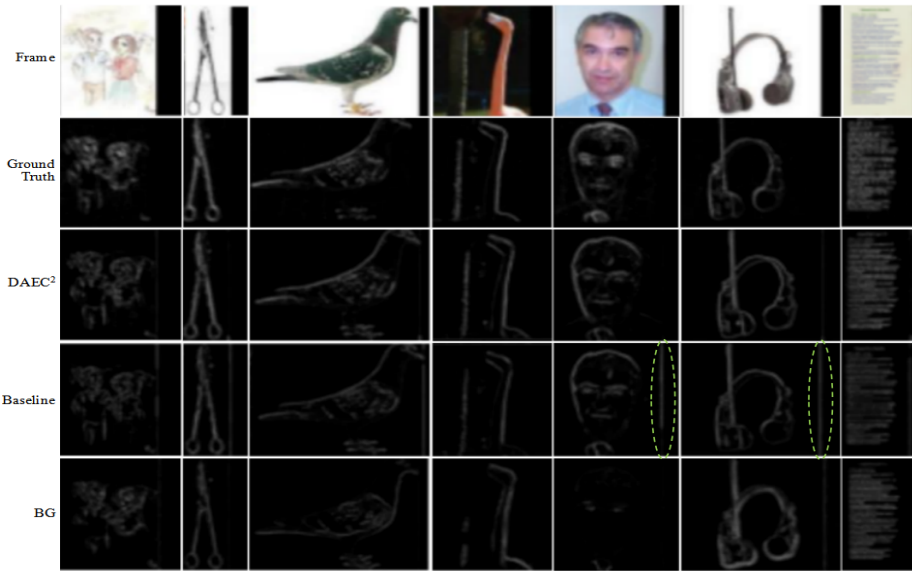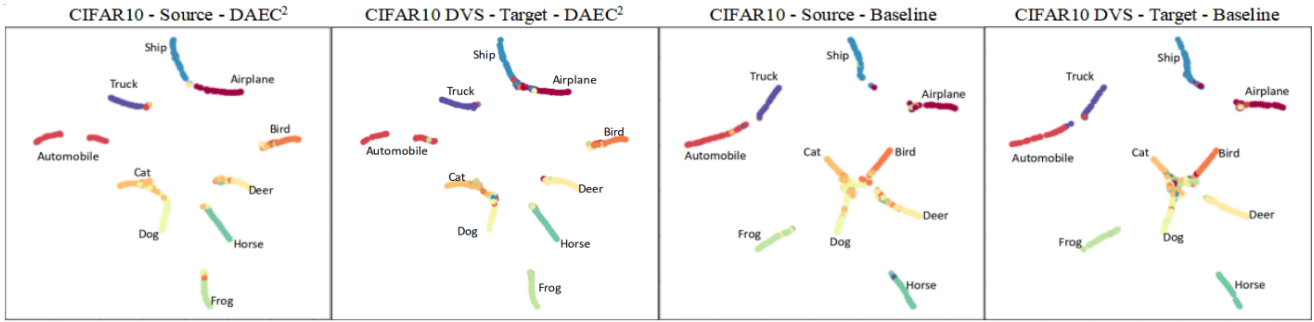

Figure 7: Images generated by three models. The input frame and event are draw from test set, and they are not paired. A random event from test set is paired with a frame, which is not shown. Green circles indicate that unnecessary lines are present in events generated by baseline model but should be discarded. These lines are boundaries of input frames, and are successfully discarded when contrastive learning loss and uncorrelated conditioning are introduced.

UMap [31] on the output of last layer of the classifier. All inputs are draw from test sets of CIFAR10 and CIFAR10-DVS. we observe that in both baseline and $DAEC^2$ is effective and the two empirical distributions are aligned; however, class clusters are more separated and have less overlap when $DAEC^2$ is used. This observation demonstrates that our algorithm improves model generalizability and leads to training a model to domain shift in the input spaces.

Effect of $DAEC^2$ on the quality of generated images: Figure 7 presents examples of generated fake events of our ground truth, $DAEC^2$, baseline, and BG[32]. The inputs are drawn from test sets of Caltech101 and N-Caltech101, and they are not paired. A close visual inspection reveals that our model leads to the best visual quality. Without our constraints, the model cannot identify and discard all useless features, i.e., the boundaries between frame images and the padding (examples shown by green circles in Figure 7). We can see such unnecessary lines in almost every generated event by the baseline model. Contrastive learning loss and uncorrelated conditioning successfully identify these unnecessary features in input frames. Moreover, the

events generated by $DAEC^2$ are clearer with fewer missing object boundaries. Thus, we can conclude that our proposed constraints help encoders to extract useful information and discard unnecessary information from the unpaired images.

6. Conclusions

We introduced contrastive learning based on augmentation - invariant representation and uncorrelated conditioning to perform UDA from frame-based data to event-based data. Our experiments demonstrate both ideas are effective and lead to improved performance on two event-based benchmarks, leading to a performance comparable with supervised learning using annotated target domain data. Uncorrelated conditioning on features extracted on the same input is an unprecedented ideas and its effectiveness in UDA motivates its adoption in other tasks as our future work. Given our empirical observations, uncorrelated conditioning can be used wherever we need to extract uncorrelated features from one same input or improving object detection.

References

[1] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE international conference on computer vision*, pages 769–776, 2013. 1

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017. 1

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019. 4

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 7

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 4

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 4

[7] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super resolve intensity images from events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2768–2776, 2020. 2

[8] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks, 2020. 6

[9] Guillermo Gallego, Christian Forster, Elias Mueggler, and Davide Scaramuzza. Event-based camera pose tracking using a generative event model. *ArXiv*, abs/1510.01972, 2015. 3

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 1

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of International Conference on Machine Learning*, pages 1180–1189, 2015. 2

[12] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3586–3595, 2020. 1, 2

[13] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data, 2019. 6

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 4, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[18] Yuhuang Hu, Tobi Delbruck, and Shih-Chii Liu. Learning to exploit multiple vision modalities by using grafted networks. In *European Conference on Computer Vision*, pages 85–101. Springer, 2020. 1

[19] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021. 1, 2

[20] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan, 2018. 4

[21] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 2

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[23] Cifar10-dvs Li. an event-stream dataset for object classification, front. 5

[24] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks, 2022. 6

[25] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A $128 \times 128120db15\mu$ s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008. 1

[26] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond, 2019. 6

[27] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of International Conference on Machine Learning*, pages 97–105, 2015. 2

[28] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018. 2

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017. 2

[30] Xin Luo, Wei Chen, Zhengfa Liang, Chen Li, and Yusong Tan. Adversarial style discrepancy minimization for unsupervised domain adaptation. *Neural Networks*, 157:216–225, 2023. 2

[31] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. 7, 8

[32] Nico Messikommer, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. 2022. 1, 2, 6, 8

[33] Manasi Muglikar, Diederik Paul Moeys, and Davide Scaramuzza. Event guided depth sensing. 2021. 6

[34] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018. 1, 2

[35] Garrick Orchard, Ajinkya Jayawant, Gregory Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades, 2015. 3, 5

[36] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2239–2247, 2019. 1

[37] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3934–3941, 2018. 2

[38] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 2

[39] Mirco Planamente, Chiara Plizzari, Marco Cannici, Marco Ciccone, Francesco Strada, Andrea Bottino, Matteo Matteucci, and Barbara Caputo. Da4event: towards bridging the sim-to-real gap for event cameras using domain adaptation. *IEEE Robotics and Automation Letters*, 6(4):6616–6623, 2021. 1, 4

[40] Bharath Ramesh, Hong Yang, Garrick Orchard, Ngoc Anh Le Thi, Shihao Zhang, and Cheng Xiang. Dart: Distribution aware retinal transform for event-based cameras, 2017. 6

[41] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 1

[42] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera, 2019. 6

[43] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2021. 1, 2

[44] Mohammad Rostami. Lifelong domain adaptation via consolidated internal distribution. *Advances in neural information processing systems*, 34:11172–11183, 2021. 2

[45] Mohammad Rostami. *Transfer Learning Through Embedding Spaces*. CRC Press, 2021. 1

[46] Mohammad Rostami. Increasing model generalizability for unsupervised visual domain adaptation. In *Conference on Lifelong Learning Agents*, pages 281–293. PMLR, 2022. 1

[47] Mohammad Rostami and Aram Galstyan. Cognitively inspired learning of incremental drifting concepts. In *International Joint Conference on Artificial Intelligence*, 2023. 1

[48] Mohammad Rostami and Aram Galstyan. Overcoming concept shift in domain-aware settings through consolidated internal distributions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 9623–9631, 2023. 2

[49] Mohammad Rostami, David Huber, and Tsai-Ching Lu. A crowdsourcing triage algorithm for geopolitical event forecasting. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 377–381. ACM, 2018. 1

[50] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018. 2

[51] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification, 2018. 6

[52] Serban Stan and Mohammad Rostami. Unsupervised model adaptation for continual semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2593–2601, 2021. 2

[53] Serban Stan and Mohammad Rostami. Domain adaptation for the segmentation of confidential medical images. In *British Machine Vision Conference*, 2022. 2

[54] Serban Stan and Mohammad Rostami. Secure domain adaptation with multiple sources. *Transactions on Machine Learning Research*, 2022. 2

[55] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 1

[56] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 2

[57] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10081–10090, 2019. 2

[58] Lin Wang, Tae-Kyun Kim, and Kuk-Jin Yoon. Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8315–8325, 2020. 2

[59] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 2

[60] Xinxing Xu, Joey Tianyi Zhou, IvorW. Tsang, Zheng Qin, Rick Siow Mong Goh, and Yong Liu. Simple and efficient learning using privileged information, 2016. 3, 5

[61] Song Zhang, Yu Zhang, Zhe Jiang, Dongqing Zou, Jimmy Ren, and Bin Zhou. Learning to see in the dark with events. In *European Conference on Computer Vision*, pages 666–682. Springer, 2020. 2

[62] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5031–5040, 2019. 2

[63] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2021. 1, 2

[64] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 3