Exploring Joint Embedding Architectures and Data Augmentations for Self-Supervised Representation Learning in Event-Based Vision

Sami Barchid[1] José Mennesson[2] Chaabane Djéraba[1]

[1] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL
[2] IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems
F-59000 Lille, France

[1]{sami.barchid, chaabane.djeraba}@univ-lille.fr
[2] jose.mennesson@imt-nord-europe.fr

Abstract

This paper proposes a self-supervised representation learning (SSRL) framework for event-based vision, which leverages various lightweight convolutional neural networks (CNNs) including 2D-, 3D-, and Spiking CNNs. The method uses a joint embedding architecture to maximize the agreement between features extracted from different views of the same event sequence. Popular event data augmentation techniques are employed to design an efficient augmentation policy for event-based SSRL, and we provide novel data augmentation methods to enhance the pretraining pipeline. Given the novelty of SSRL for event-based vision, we elaborate standard evaluation protocols and use them to evaluate our approach. Our study demonstrates that pretrained CNNs acquire effective and transferable features, enabling them to achieve competitive performance in object or action recognition across various commonly used event-based datasets, even in a low-data regime. This paper also conducts an experimental analysis of the extracted features regarding the Uniformity-Tolerance tradeoff to assess their quality, and measure the similarity of representations using linear Center Kernel Alignement. These quantitative measurements reinforce our observations from the performance benchmarks and show substantial differences between the learned representations of all types of CNNs despite being optimized with the same approach.

1. Introduction

Event cameras [13] are emergent visual sensors that operate on a fundamentally different principle than traditional frame-based cameras. Instead of capturing frames at a fixed rate, event cameras asynchronously measure brightness changes at the pixel level and output a stream of related events that efficiently encode the spatiotemporal dynamics of the scene. These bio-inspired sensors have several advantages over conventional cameras, such as lower latency (at a microsecond level), higher dynamic range, sparser representation than frames, and better energy efficiency. Thanks to these advantages, event cameras draw a lot of attention recently, notably for computer vision applications for neuromorphic vision systems [10], autonomous navigation [23], etc.

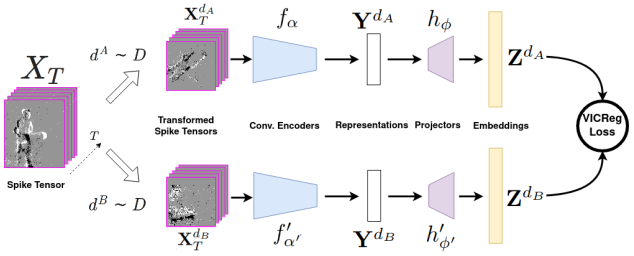

Figure 1. Overview of the proposed event-based SSRL framework based on a joint embedding architecture.

Computer vision solutions that process event streams constitute a new domain known as "Event-based vision". A substantial part of this field leverages image-like inputs from event accumulation, and adapts popular frame-based vision approaches, such as deep Convolutional Neural Networks (CNNs), to process them. On the other hand, recent advances in training bio-inspired neurons [32] have enabled the development of deep learning techniques using Spiking Neural Networks (SNNs) [2, 12], which are inherently capable of processing asynchronous events. These numerous techniques achieve state-of-the-art performance but commonly necessitate substantial labeled data, similar to frame-based vision models. Currently, there are few large-scale

event datasets available [22], which emphasizes the necessity to mitigate the dependence on labeled event data.

In frame-based vision, Self-Supervised Representation Learning (SSRL) has made great progress recently to pretrain deep learning models for finetuning downstream tasks. State-of-the-art SSRL methods [5, 43] utilize data augmentation on unlabeled images to learn features without supervision. In this work, we propose to adapt this strategy for event cameras by defining a simple event-based SSRL framework based on a joint embedding architecture [5]. Given the novelty of event-based SSRL, we formulate experimental protocols to evaluate this model and future works, on both object recognition (*i.e*. short event sequences) and human activity recognition (*i.e*. longer sequences with challenging spatiotemporal information). In addition, we use our simple framework as a context to compare data augmentation techniques specific to events, known as Event Data Augmentations (EDAs). Using various lightweight Convolutional Encoders (ConvEncs), namely 2D-, 3D- and Spiking CNNs, our method achieves competitive or superior performance in downstream tasks compared to state-of-the-art techniques, while using lighter neural networks and/or fewer unlabeled data.

Our contributions can be summarized as follows:

- We introduce an SSRL framework for event-based vision (shown in Fig. 1) that can be used with various types of ConvEncs, including 2D-, 3D-, and Spiking CNNs. Pretrained models using this approach achieve competitive performance on downstream classification tasks.
- We assess prevalent EDAs for giving insight into good augmentation policy design in event-based SSRL and introduce novel techniques to improve the method.
- We establish various evaluation protocols on downstream tasks to assess the effectiveness of event-based SSRL methods.
- We conduct an empirical study on the representations learned by the resulting pretrained ConvEncs. We evaluate their quality using a Uniformity-Tolerance analysis [18] and measure their similarity using the linear CKA analysis [25].

2. Related Works

2.1. Self-Supervised Representation Learning

In frame-based vision, several strategies exist for SSRL, mainly depending on a defined pretext task employed to learn useful features (*e.g*. masked modeling [8], rotation parameters prediction [14], ...). Popular strategies utilize a distribution of data augmentation to create diverse views of images for learning invariant features, but these pretext tasks can lead to trivial solutions (*i.e*. collapse), so SSRL approaches differ in the method to address this issue. Contrastive Learning trains a model to attract positive pairs (i.e., augmented views of a given image) and repel negative pairs (i.e., augmented views of two different images) [9, 19]. Clustering methods [7] involve grouping augmented views into clusters as pseudo-labels for representation learning. More recently, it has been shown that simpler approaches based on joint embedding architectures and covariance regularization can avoid collapsing solutions (*e.g*. Barlow Twins [43], VICReg [5]). These methods are easier to use, do not rely on sophisticated bag-of-tricks [16], and achieve state-of-the-art performance, which makes them a preferred candidate to explore event-based SSRL for convolutional neural networks.

2.2. Addressing Scarcity of Event-based Datasets

Given the relative novelty of event-based vision, few public datasets exist for a given task, and most of these datasets have a limited number of labeled data. Pretraining a model on larger datasets is a common technique to improve the performance of downstream tasks with limited labeled data. Many strategies exist to obtain efficient pretraining event datasets. [22, 33] place an event camera in front of a screen to record frame-based datasets so as to obtain simulated event streams, but it requires a calibrated hardware setup. Video-to-events converters [21, 36] can achieve the same event stream simulation more easily, and can, to some extent, simulate real-world noisy conditions of neuromorphic sensors. Rather than converting frame-based datasets, self-supervised learning has been used in low-level event-based vision tasks, such as optical flow for frames+events [44], or frame reconstruction with a generative model [34]. More recently, event-based SSRL methods were designed for deep learning models: [27] investigates the applicability of SimCLR [9] for SNNs with a novel event data augmentation policy. [24] proposes a masked modeling strategy for pretraining large-scale Vision Transformers on event data. Similarly, [42] designs an event-based SSRL framework for Vision Transformers using both events and RGB frames. While these pioneering works show great results on event object recognition with short event sequences, longer spatiotemporal vision tasks, such as human activity recognition [1, 28], are still not investigated with event-based SSRL. In comparison, our approach, primarily focused on lightweight convolutional encoders (2D, 3D, and Spiking), shows strong performance on both short and long event sequences (object recognition and human activity recognition, respectively) with fewer unlabeled data for pretraining.

2.3. Event Data Augmentations

Data augmentation plays a crucial role in training deep learning models by promoting invariant feature learning, which improves generalization and overall performance. Prior works [27] on Event Data Augmentation (EDA) aim to adapt popular frame-based augmentations to the event domain, such as Flipping, Rolling, Rotation, CutMix, etc. Additionally, techniques like EventDrop [17], which drops events either randomly, spatially, or temporally, and EventMix [37], which combines different event streams using a random spatiotemporal mask, efficiently consider the temporal dimension of event sequences. [31] proposes temporal shifting of events specifically for contact force estimation. The Tonic library [26] offers common EDAs, such as time reversal, polarity flip, uniform background noise, among others. As many SSRL approaches rely on input sample transformations, these EDAs are suitable candidates for efficiently distorting input event streams. We select relevant EDAs to explore good design practices for distortion distribution in event-based SSRL and propose additional EDA techniques to enhance our approach.

3. Method

This section presents the details of the proposed framework for event-based SSRL, which leverages various data augmentation techniques. First, the problem formulation is introduced, followed by an explanation of the joint embedding architecture that is optimized using the state-of-the-art VICReg loss [5]. Two variants of the architecture are described: the first one, named "Twins" employs a siamese-like network structure, while the second one, named "Student-Teacher," adopts an asymmetric structure. Subsequently, the section explains the data augmentation techniques used with our SSRL approach. While there are some data augmentation techniques in previous works, we propose novel ones that are specifically designed for event-based vision.

3.1. Preliminary Notions

An event camera of resolution $H \times W$ produces N asynchronous events $\mathcal{E} = \{e_i\}_{i=1}^{N}$ during a time interval $\Delta_{\mathcal{T}}$. Each event e_i is represented by a tuple of 4 values: $e_i = (x_i, y_i, t_i, p_i)$, where (x_i, y_i) are the pixel coordinates, t_i is the timestamp, and $p_i \in \{1, -1\}$ denotes the sign of the polarity change.

To address the unsuitability of the asynchronous nature of events for various computer vision techniques [13], a commonly used method for event representation is to discretize a stream of events $\mathcal{E}$ into a sequence of T binary event frames $\mathbf{X}_T \in \mathbb{B}^{T\times 2\times H\times W} = \{X_t\}_{t=1}^{T}$. This is achieved by accumulating events in T successive time intervals $\frac{\Delta_{\mathcal{T}}}{T}$, resulting in the sequence of binary frames and thus the corresponding spike tensor $\mathbf{X}_T$.

In this work, an Event Data Augmentation (EDA) process can be formulated as a function $d(\cdot)$ that takes a given spike tensor $\mathbf{X}_T$ as input and returns a distorted version $\mathbf{X}_T^d \in \mathbb{B}^{T\times 2\times H\times W}$, such that $\mathbf{X}_T^d = d(\mathbf{X}_T)$. The function $d(\cdot)$ can be either one single augmentation technique (*e.g.* a random translation) or a composition of several functions $d = d_1 \circ d_2 \circ \dots$.

3.2. Joint Embedding Architecture

Our method is illustrated in Fig. 1. Similarly to popular trends of SSRL [5, 43], it is based on a joint embedding architecture, where two neural networks (here, two convolutional encoders) are trained to infer similar embeddings from two different views of the same input.

Formally, two EDA functions d_A and d_B are sampled from a distribution of transformations D. Given a spike tensor $\mathbf{X}_T$, each function creates a distorted view, *i.e.* $\mathbf{X}_T^{d_A} = d_A(\mathbf{X}_T)$ and $\mathbf{X}_T^{d_B} = d_B(\mathbf{X}_T)$. These two views are encoded using two functions $f_\alpha(\cdot)$ and $f'_{\alpha'}(\cdot)$ to obtain two *representations* $\mathbf{Y}^{d_A}, \mathbf{Y}^{d_B} \in \mathbb{R}^C$ given by $\mathbf{Y}^{d_A} = f_\alpha(\mathbf{X}_T^{d_A})$ and $\mathbf{Y}^{d_B} = f'_{\alpha'}(\mathbf{X}_T^{d_B})$. The functions f_α and $f'_{\alpha'}$ are typically convolutional encoders (ConvEncs) with parameters α and α', respectively. Note that the representations are features computed over the whole sequence and thus lose the temporal dimension discretized by T time-steps. Finally, the representations are expanded using two *projectors* h_ϕ and $h'_{\phi'}$ to produce the two *embeddings* $\mathbf{Z}^{d_A} = h_\phi(\mathbf{Y}^{d_A})$ and $\mathbf{Z}^{d_B} = h'_{\phi'}(\mathbf{Y}^{d_B})$, where $\mathbf{Z}^{d_A}, \mathbf{Z}^{d_B} \in \mathbb{R}^{3C}$. The projectors are a neural networks composed of three linear layers, each with $3C$ output units, and the first two layers are followed by BatchNorm and ReLU. The whole framework is optimized using the VICReg loss on two batches of embeddings (see details of VICReg in [5]).

3.2.1 Convolutional Encoders

In our SSRL framework, we evaluate three types of commonly used ConvEncs in event-based vision: 2D-CNN, 3D-CNN, and CSNN (*i.e.* Convolutional Spiking Neural Network). To ensure a fair comparison, we use a lightweight ResNet-like architecture for all three encoders, namely ResNet-18 [20], MC3-ResNet-18 [39], and SEW-ResNet-18 [12] for 2D-CNN, 3D-CNN, and CSNN, respectively. To accommodate these different types of ConvEncs, small clarifications on the model or minor modifications of the original formulation provided in Sec. 3.1 are necessary for the 2D-CNN and CSNN.

2D-CNN To handle the temporal dimension of spike tensors with 2D-CNNs, we concatenate the temporal dimen-

sion with the channel dimension, which results in a spike tensor of size $\mathbf{X}_T \in \mathbb{B}^{TC \times H \times W}$, similar to [3, 4].

CSNN Instead of artificial neurons, CSNNs are composed of spiking neurons that natively have the capability of processing the temporal dimension of spike tensors. In this work, we employ the Integrate-and-Fire (IF) neuron model [30], which integrates input spikes weighted by synaptic weights into an internal state called "membrane potential". When the membrane potential of a spiking neuron surpasses a defined value (*i.e.* the "threshold"), an output spike is emitted and the potential is reset to 0. The discrete dynamics of a layer l of IF neurons at a specific time-step $1 \leq t \leq T$ is described as follows:

$$U_t^l = U_{t-1}^l + \mathcal{W}^l X_{t-1}^{l-1} - \theta X_t^l \quad (1)$$

$$X_t^l = \Theta(U_t^l - \theta) \quad (2)$$

where U_t^l denotes the membrane potentials in the layer, $\mathcal{W}^l$ is the set of synaptic weights, $X_t^l \in \mathbb{B}$ denotes the output spike tensor. X_t^l consists of 1's when the related element of U_t^l exceeds the threshold value θ, and 0's otherwise. In this work, the threshold is set to 1 in the whole network (i.e., $\theta = 1$). This mechanism is known as the Heaviside step function ($\Theta(\cdot)$) and is formulated in Eq. (2). The whole CSNN is trained using Surrogate Gradient Learning [32].

A given CSNN outputs a sequence of binary spiking features at each time-step, *i.e.* $f_\alpha(\mathbf{X}_T) = \{Y_t^d\}_{t=1}^T$ with $Y_t^d \in \mathbb{B}^C$. Thus, it cannot be used "as is" to obtain the real-valued representation vector $\mathbf{Y}^d$. Similarly to previous works on SNNs [4], an output accumulator module on top of the CSNN is used to accumulate the spiking features Y_t^d across all time-steps and obtain the representation $\mathbf{Y}^d$.

3.2.2 Variants

We define two variants of the joint embedding architecture of our SSRL framework.

Twins: it is the standard design of joint embedding architectures, where the two ConvEncs $f_\alpha(\cdot)$ and $f'_{\alpha'}(\cdot)$ are the same architecture with shared weights, *i.e.* $\alpha = \alpha'$.

Student-Teacher: spiking neurons in CSNNs only exchange binary values (*i.e.* spikes), which are less expressive than real values processed by 2D/3D-CNNs. To address this issue, we leverage the ability of VICReg [5] to train asymmetric joint networks. The first branch $f_\alpha(\cdot)$ of the architecture is a CSNN (*i.e.* the student). The second branch $f'_{\alpha'}(\cdot)$ is a 2D/3D CNN (*i.e.* the "teacher") whose objective is to enhance the training of the CSNN.

3.3. Event Data Augmentation

Defining an effective distribution D of EDAs is critical for effective pretraining in our SSRL framework. In this section, we describe the investigated EDAs from previous works with their related random parameters. Furthermore, we propose novel techniques to extend the field for SSRL. Investigated EDAs are separated into 3 groups: Common EDAs, Drop-based EDAs, and Geometric EDAs. Visualizations of the EDAs are available in the Supplementary Materials.

Common EDAs refer to a set of commonly used distortions in event-based vision that share no specific characteristics among them. All these EDAs are made available in the Tonic [26] library.

- **Noise**: approximated background activity noise of event sensors. A random percentage $r_{\text{noise}} \in [0.5, 20]$ of uniformly distributed noisy events is added to the input event sequence.
- **Flip Polarity** (PolFlip): flips the polarity of all events, *i.e.* $p_i = -p_i$.
- **Crop**: random resized crop to a random aspect ratio $r_{\text{crop}} \in [0.08, 1.0]$.

Drop-Based EDAs refer to data augmentation where events are removed from the original spike tensor. Several drop-based primitives are defined in [17]:

- **Drop By Area** (or **Cutout**): removes events located in a random box region of dimension $r_{\text{cut}} H \times r_{\text{cut}} W$, with the parameter $r_{\text{cut}} \in [0.05, 0.3]$.
- **Drop By Time**: removes all events from the original event stream $\mathcal{E}$ in a random time interval of duration $r_{\text{time}} \times \Delta_\mathcal{T}$ with $r_{\text{time}} \in [0.1, 0.9]$
- **Random Drop**: each event has a probability $r_{\text{drop}} \in [0.1, 0.9]$ to be removed from the input event stream $\mathcal{E}$.

EventDrop [17] employs a `OneOf` policy that randomly applies one of the three EDA primitives with equal probability.

In this work, we propose an EDA primitive named **Event-Copy**: events located in a random box region of dimension $r_{\text{copy}} H \times r_{\text{copy}} W$ are *copied* to another location in the spike tensor, with $r_{\text{copy}} \in [0.05, 0.3]$.

Moreover, we extend EventDrop and propose **Event-CopyDrop**, which is essentially a `OneOf` policy using the 4 discussed drop-based primitives.

Geometric EDAs refer to techniques that transform an input event sequence spatially and are adaptations of common geometric augmentations from frame-based vision [27].

Type	Name	Probability	Parameters
Common	**Noise**	0.5	$r_{\text{noise}} \in [0.5, 20]$
	PolFlip	0.2	-
	Crop	1.0	$r_{\text{crop}} \in [0.08, 1.0]$
Geometric	**StatTran**	0.5	$r_y \in [\pm 0.2 \times H]$, $r_x \in [\pm 0.2 \times W]$
	StatRot	0.5	$r_{\text{degrees}} \in [-75, 75]$
	DynTran (ours)	0.5	$r_y \in [\pm 0.2 \times H]$, $r_x \in [\pm 0.2 \times W]$
	DynRot (ours)	0.5	$r_{\text{degrees}} \in [-75, 75]$
	StatDynGeo (ours)	0.8	$r_y \in [\pm 0.2 \times H]$, $r_x \in [\pm 0.2 \times W]$, $r_{\text{degrees}} \in [-75, 75]$
Drop-Based	**Cutout**	0.3	$r_{\text{cut}} \in [0.05, 0.3]$
	Event-Copy (ours)	0.5	$r_{\text{copy}} \in [0.05, 0.3]$
	EventDrop	0.75	$r_{\text{cut}} \in [0.05, 0.3]$, $r_{\text{time}} \in [0.1, 0.9]$, $r_{\text{drop}} \in [0.1, 0.9]$
	EventCopyDrop (ours)	0.8	$r_{\text{cut}} \in [0.05, 0.3]$, $r_{\text{time}} \in [0.1, 0.9]$, $r_{\text{drop}} \in [0.1, 0.9]$, $r_{\text{copy}} \in [0.05, 0.3]$

Table 1. **Summary of investigated EDAs**, including their related probability in the distribution D, and their random parameters.

- **Static Translation** (StatTran): translates the whole input spike tensor vertically by $r_y \in [\pm 0.2 \times H]$ and horizontally by $r_x \in [\pm 0.2 \times W]$.
- **Static Rotation** (StatRot): rotates whole spike tensor to a random angle $r_{degrees} \in [-75, 75]$.

We propose novel geometric transformations that also augment the event sequences temporally:

- **Dynamic Translation** (DynTran): operates a *linearly progressive* translation from 0 to $r_y \in [\pm 0.2 \times H]$ vertically and from 0 to $r_x \in [\pm 0.2 \times W]$ horizontally along the time axis.
- **Dynamic Rotation** (DynRot): *progressively* rotates the whole spike tensor from 0 to a random angle $r_{degrees} \in [-75, 75]$ along the time axis.
- **StatDynGeo**: applies a `OneOf` policy with the four mentioned geometric EDAs: [StatTran, StatRot, DynTran, DynRot].

3.3.1 EDA Distribution Policy

At each sampling from the distribution D (*i.e.*, when transforming the input events during a training step), each selected EDA has a given probability to be employed for composing the EDA function d_A or d_B. Tab. 1 summarizes the probability scores for all investigated EDAs, as well as their random parameters.

4. Experimental Methodology

In this section, we outline the methodology used to evaluate the performance of our event-based SSRL baseline and quantitatively analyze the extracted features of all pretrained ConvEncs. The experiments conducted involve: **(1)** evaluating downstream task performance on event-based recognition benchmarks ; **(2)** comparing the extracted representations of ConvEncs through Uniformity-Tolerance analysis [18] ; and **(3)** measuring the similarity of extracted features from all ConvEncs using linear Center Kernel Alignment (CKA). Implementation details are given at the end of the section.

Dataset	Resolution	Duration	#Samples Total	Train	Val.	Classes
DVSGesture [1]	128×128	$\pm$6s	1342	1078	264	11
DailyAction-DVS [28]	346×260	$\pm$5s	1440	1152	288	12
ASL-DVS [6]	240×180	$\pm$0.1s	100800	80640	20160	24
N-Cars [38]	304×240	$\pm$0.1s	24029	15422	8607	2
NCaltech-101 [33]	Variable	$\pm$0.3s	8709	7838	871	101

Table 2. **Summary of the studied datasets**.

4.1. Datasets Investigated

Our approach differs from prior works [24, 42] by evaluating the ability of the methods to be pretrained with limited data, emphasizing faster convergence without relying on large-scale datasets such as N-ImageNet [22]. We select two types of event-based recognition datasets: **(1)** short-length sequences for object recognition [6, 33] (similar to other works), and **(2)** human action recognition datasets. The former is widely available in the state-of-the-art, whereas the latter is more challenging as it relies on the temporal dimension. Table 2 summarizes the pertinent details of the evaluated datasets.

4.2. Performance Benchmarks

We formulate standard protocols to evaluate event-based SSRL methods on their performance (*i.e.* accuracy scores) on downstream tasks.

Linear Evaluation Protocol A linear classifier is trained on the fixed representations obtained from pretraining one of the ConvEncs with our method using the train set of an event-based dataset. The achieved top-1 accuracy on the validation set of the same dataset is reported. The objective of this evaluation protocol is to show the ability of an event-based SSRL method to extract useful unsupervised features. We employ three datasets of different sizes: DVSGesture [1], N-Caltech101 [33] and ASL-DVS [6].

Semi-Supervised Training The objective of this benchmark is to evaluate the ability of a given SSRL approach to reduce the need for labeled data. After pretraining on a given dataset with our SSRL framework, we finetune the ConvEnc using a labeled subset of the same dataset (*i.e.* using a defined percentage of the original training labels). The accuracy score obtained on the validation set is reported. Evaluations are conducted on DVSGesture [1], N-Caltech101 [33], and ASL-DVS [6].

Transferring to Another Dataset This evaluation protocol aims at evaluating the capacity to transfer the features learned from one dataset to another. Specifically, we evaluate the transfer learning performance of a pretrained ConvEnc by training a linear classifier on another dataset while keeping the ConvEnc's parameters fixed. We report the top-1 accuracy on the validation set. We define two transfer learning scenarios: DVSGesture [1] (pretraining) to DailyAction-DVS [28] for activity recognition, and ASL-DVS [6] (pretraining) to N-Cars [38] for object recognition on short-length sequences.

4.3. Uniformity-Tolerance Analysis

In evaluating the quality of the learned representations, we utilize Uniformity and Tolerance metrics [40, 41]. The Uniformity metric $\mathcal{L}_{\text{uni}}$ measures the proximity of the representations to a uniform distribution on the feature hypersphere, indicating that the ConvEnc learns separable representations. The Tolerance metric $\mathcal{L}_{\text{tol}}$ utilizes ground truth labels to assess the extent to which the representations capture the semantic relationships between the samples. Uniformity and Tolerance are respectively given in Eq. (3) and Eq. (4).

$$\mathcal{L}_{\text{uni}} = \log \mathop{\mathbb{E}}_{x,y \sim p_{data}} [e^{-t||f(x)-f(y)||_2^2}] \tag{3}$$

$$\mathcal{L}_{\text{tol}} = \mathop{\mathbb{E}}_{x,y \sim p_{data}} [(||f(x)||_2^T ||f(y)||_2) \cdot I_{gt(x)=gt(y)}] \tag{4}$$

, where $f(\cdot)$ denotes a ConvEnc that extracts representations from a sample in the dataset p_{data}, $I_{gt(x)=gt(y)}$ denotes the indicator function used to identify whether any given pair of samples x and y share the same label, with 1 indicating a match ($gt(x) = gt(y)$) and 0 indicating a mismatch. A scaling hyper-parameter $t = 2$ is employed.

Prior research [18, 40] suggests that SSRL approaches achieve optimal representation quality by balancing Uniformity and Tolerance metrics. Our objective is to compare pretrained ConvEncs based on this trade-off.

4.4. Linear CKA for Similarity Assessment

Following similar studies in frame-based SSRL [18], we compute the linear Centered Kernel Alignement (CKA) to assess the similarity of representations extracted by various pretrained ConvEncs. To do so, we obtain the representation matrices for two different ConvEncs (*e.g.* 2D-CNN and CSNN), denoted A and B, and compute their Gram matrices as $K = AA^T$ and $L = BB^T$. The CKA value is calculated as the normalized Hilbert-Schmidt Independence Criterion (HSIC) [15]:

$$\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K)\text{HSIC}(L, L)}} \tag{5}$$

4.5. Implementation Details

All experiments are implemented with PyTorch [35] (+ SpikingJelly [11] to simulate spiking neurons) and run on an NVIDIA A40 GPU. For all experiments, the models are trained using an SGD optimizer with a learning rate of 0.01, and a cosine annealing scheduler [29]. Every sample is resized to a resolution of 128×128. As for the VICReg loss function, we use the same coefficients as the original paper [5]. Spike tensors are generated with $T = 12$ timesteps. Code is available on `https://github.com/Barchid/exploring_event_ssl`.

5. Analysis

5.1. Investigation on Event Data Augmentations

First and foremost, we evaluate the impacts of the discussed EDAs on the overall performance of all ConvEncs in order to find a good distribution D for subsequent experiments and potential future works. The EDAs discussed in Sec. 3.3 are evaluated following the Linear Evaluation Protocol on DVSGesture [1] (see Sec. 4.2). We divide this investigation into three incremental stages, corresponding to the three types of EDAs described in Sec. 3.3: **(1)** Common EDAs; **(2)** Geometric EDAs; and **(3)** Drop-based EDAs. For each stage, we keep the best-performing EDA configuration from the previous stage to evaluate the EDAs of the current stage. Table 3 reports the results of the study.

Generally, we observe that CSNNs have the worst performance by a large margin, but this issue is greatly mitigated by the Student-Teacher variants, which can lead to promising future directions specific to SNNs.

In Stage 1, we incrementally incorporate all common EDAs and observe that the models exhibit better performance with an increasing number of EDAs. Combining the three transformations ranks first or second in performance for almost every ConvEnc. The improved performance with increasing EDAs can be attributed to their non-overlapping nature, which augments the original sequences without sacrificing semantic information.

In Stage 2, we compare static and dynamic translations/rotations, observing that ConvEncs with spatiotemporal processing capabilities (*i.e.* CSNN and 3D-CNN) perform better with dynamic transformations. The proposed StatDynGeo achieves most of the best results, highlighting the efficacy of `OneOf` policies for event-based SSRL. Surprisingly, geometric distortions are found to harm SSRL for 2D-CNNs compared to only applying Common EDAs.

In Stage 3, we compare all EDAs and observe the benefits of incorporating one drop-based EDA in the transformation distribution. While all drop-based transformations report at least one top performance for a specific ConvEnc, EventCopyDrop repeatedly performs well (second or first), which makes it a strong contender. Therefore, adding

Figure 2. **Linear CKA of all discussed ConvEncs** for the initial convolutional layer (Conv. 1), the first three residual blocks of the ResNet architectures (Res. 2/3/4), and the output representations. **3D**, **2D**, and **S** are **3D**-CNN, **2D**-CNN and CSNN respectively. $\mathbf{S_{2D}}$ and $\mathbf{S_{3D}}$ refers to the CSNN coupled with 2D/3D-CNN in the Student-Teacher variant. The subscript $_{\mathbf{sup}}$ refers to "supervised learning" of the related ConvEnc.

Stage 1: Common EDAs			Twins			Student-Teacher			
Noise	Crop	PolFlip	CSNN	2D	3D	CSNN	2D	CSNN	3D
			12.12	11.36	57.95	59.47	61.36	58.33	55.68
✓			**60.23**	74.24	69.70	62.12	64.02	68.26	60.23
✓			56.44	81.82	74.62	**73.86**	**76.89**	69.32	64.02
✓	✓		56.44	**83.33**	**76.52**	73.11	74.62	**70.63**	**67.80**
✓	✓	✓	56.44	**83.33**	**76.52**	73.11	74.62	**70.63**	**67.80**

Stage 2: Geometric EDAs					Twins			Student-Teacher			
StatTran	StatRot	DynTran	DynRot	StatDynGeo	CSNN	2D	3D	CSNN	2D	CSNN	3D
✓	✓				49.24	77.65	65.15	60.98	68.18	65.53	62.5
		✓	✓		68.56	75.38	76.14	**74.24**	**73.11**	**72.73**	67.05
				✓	**68.94**	**79.55**	**77.27**	67.05	70.45	71.97	**68.18**

Stage 3: Drop-based EDAs				Twins			Student-Teacher			
Cutout	EventDrop	Event-Copy	EventCopyDrop	CSNN	2D	3D	CSNN	2D	CSNN	3D
✓				**71.59**	85.11	88.64	71.59	72.35	68.56	65.91
	✓			68.18	**87.12**	75	75	76.52	**76.52**	**76.14**
		✓		65.16	81.82	83.33	76.52	**77.52**	73.86	68.94
			✓	70.83	**87.12**	**89.39**	**76.89**	75.76	75	73.48

Table 3. **Investigation on EDAs:** accuracy scores obtained on the linear evaluation protocol with DVSGesture [1].

Event-Copy in the `OneOf` relationship of EventDrop [17] is a recommended choice for defining a good and non-specific distribution D.

To summarize, we can draw three rules of thumb from this study for designing a generally effective distribution D: **(1)** incorporating more common EDAs results in better pre-training performance ; **(2)** selecting one geometric EDA and one drop-based EDA strongly enhances the SSRL pipeline, but ConvEncs with spatiotemporal processing perform better with dynamic geometric transformations; and **(3)** `OneOf` policies perform well by enabling the use of several similar EDAs without overlapping.

For the rest of this paper, we employ the distribution D = {Noise, Crop, PolFlip, StatDynGeo, EventCopyDrop}. The results obtained by the teacher networks are no longer reported in the remainder of this section, as they do not benefit from the asymmetric design compared to the Twins variants.

5.2. Performance Benchmarks

Table 4 reports the results of the linear evaluation and the semi-supervised training protocols (described in Sec. 4.2). Our approach achieves encouraging performance across all datasets, demonstrating the effectiveness of joint embedding architectures for event-based SSRL. The promising results obtained on Semi-Supervised Training also suggest that our approach successfully reduces the reliance on labeled data. We observe that 2D-CNN and 3D-CNN outperform CSNNs due to the binarized activation function of the spiking mechanism, which is less expressive than real-valued activations. This finding suggests that specialized SSRL methods for spiking neurons could better suit CSNNs, even if the Student-Teacher variant mitigates the lower results.

Dataset	Protocol	CSNN	2D	3D	CSNN_{2D}	CSNN_{3D}
DVSGesture	Linear	70.83	87.12	**89.39**	76.89	76.52
	SemiSup-10%	60.98	75.52	**81.44**	66.67	69.31
	SemiSup-25%	75.00	87.12	**90.15**	76.14	80.30
N-Caltech101	Linear	64.29	64.39	**69.46**	62.34	65.67
	SemiSup-10%	56.72	**64.64**	62.80	53.96	53.50
	SemiSup-25%	66.02	**72.79**	71.64	62.22	59.93
ASL-DVS	Linear	95.32	**99.38**	98.68	97.87	97.30
	SemiSup-05%	95.66	**97.06**	96.62	93.54	95.66
	SemiSup-10%	99.51	99.64	**99.70**	99.48	99.48

Table 4. **Results of Linear Evaluation and Semi-Supervised Training**. "SemiSup-XX%" denotes the semi-supervised training protocol where XX% of the training set is used for fine-tuning. CSNN_{2D} and CSNN_{3D} are CSNNs pretrained using the Student-Teacher variant with a 2D-CNN and a 3D-CNN, respectively.

Datasets						
Pretrain	Linear	CSNN	2D	3D	CSNN_{2D}	CSNN_{3D}
DVSGesture	**DailyAction-DVS**	77.93	88.28	84.83	**91.03**	87.59
ASL-DVS	**N-CARS**	92.81	94.61	**95.64**	93.30	93.35

Table 5. **Results of the Transfer Learning Protocol.**

Table 5 reports the performance obtained in the transfer learning protocol from one dataset to another. The reported performance demonstrates the transferability of the features learned by our approach and hence verifies the efficiency in pretraining event-based vision models.

To put our results in perspective, we report the best results obtained in our experiments and compare them with

fully supervised methods from the state-of-the-art (reported in Appendix B in the supplementary materials). We show that our ConvEncs, without finetuning on the whole training set, achieve competitive performance against heavier models and even achieve top performance (notably on DailyAction-DVS [28]). Consequently, we demonstrate that our SSRL framework can be a competitive and cost-effective alternative to large-scale supervised learning for future works.

As for other works on event-based SSRL [24, 42], our method can primarily be compared to performance on N-Cars. Specifically, while [24] and [42] achieve 98.55% and 97.93% accuracy, respectively, with heavier Vision Transformer models and full supervision, our lightweight ConvEnc (here, 3D-CNN) achieves a lower but promising accuracy of 95.64%. However, direct comparison is not straightforward since these works fine-tune on the whole N-Cars train set, while our transfer learning protocol only trains the linear classifier, and the ConvEncs are only pretrained on ASL-DVS. In addition, our method has a more general scope as it investigates longer event sequences such as human activity recognition.

Our method's competitive performance with lightweight ConvEncs suggests that event-based SSRL can learn good representations, enabling exploration of diverse datasets and tasks instead of relying on large and expensive datasets. This potential for event-based vision to expand to innovative applications opens up new possibilities.

5.3. Uniformity-Tolerance Analysis

Figure 3 shows the uniformity and tolerance of all ConvEncs pretrained on DVSGesture, and ASL-DVS. In all datasets, we find important differences between all pretrained ConvEncs, suggesting that our approach does not impact all convolutional architectures equally, even when optimized in the same manner. Interestingly, we observe strong unbalanced results for 2D/3D-CNNs on short sequence datasets (*i.e.* ASL-DVS [6]), while these ConvEncs obtain balanced results on spatiotemporal activity recognition datasets [1]. The strong unbalance in favor of tolerance suggests that the assumption of the Uniformity-Tolerance tradeoff [41] does not prevail for better downstream performance in event-based SSRL. CSNNs, on the other hand, show some balanced measurements, but we note an increase in tolerance with the Student-Teacher variants, which verifies the enhanced performance on downstream tasks.

5.4. Analysis on Representations Similarity

We measure the Linear CKA values of all pretrained ConvEncs on DVSGesture [1], as well as the ConvEncs trained with supervision (for comparison purposes). In addition to the final representations, we compare the features at the end of all residual blocks of the ResNet architectures.

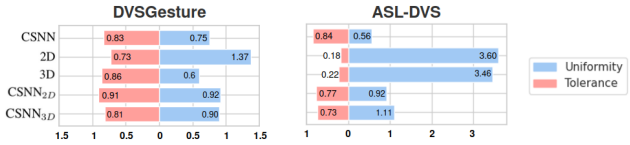

Figure 3. **Uniformity-Tolerance Analysis**.

Figure 2 shows the results of this analysis.

We observe substantial similarities with previous studies on frame-based SSRL [18]: the features extracted tend to diverge increasingly as one goes deeper into the layer, and the final representations show the most dissimilarities. In the low-level layers (Conv 1 and Res. 2), we observe dissimilarities between 2D/3D-CNNs and CSNNs, which can be explained by the differences between spiking and artificial neurons. On the other hand, we verify the findings that supervised and unsupervised representations diverge the most in the last layer.

When focusing on the analysis of representations only, we find that the Student-Teacher variants make CSNNs learn features that diverge from Twins-CSNN, but also from the Teacher ConvEncs. While the differences between the Student and the Teacher can be explained by the differences between spiking and artificial neurons, the divergence observed between all CSNNs verifies the impacts of the Student-Teacher variants on training SNNs. Interestingly, representations from CSNNs are not especially closer to 3D-CNNs than 2D-CNNS, despite their common abilities to process spatiotemporal data.

6. Conclusion

In this paper, we present a straightforward and efficient framework for event-based Self-Supervised Representation Learning (SSRL) using different types of convolutional encoders (ConvEncs), including 2D-/3D-CNNs and CSNNs in a joint embedding architecture. Our formulated evaluation protocols demonstrate that our method, without full supervision, achieves competitive performance on popular datasets (both object and activity recognition). We also investigate popular event data augmentation (EDA) techniques and introduce new methods to aid in designing a good EDA distribution policy for future works on event-based SSRL. Beyond performance benchmarks, we quantitatively evaluate the learned representations of all ConvEncs using Uniformity, Tolerance, and Linear CKA, and find substantial differences depending on the variant and type of CNN, with the Student-Teacher design proving beneficial for CSNNs. Overall, our framework can reduce the need for large-scale labeled datasets, opening up opportunities for expanding event-based vision tasks.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252, 2017. 2, 5, 6, 7, 8

[2] Sami Barchid, José Mennesson, and Chaabane Djéraba. Deep spiking convolutional neural network for single object localization based on deep continuous local learning. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–5. IEEE, 2021. 1

[3] Sami Barchid, José Mennesson, and Chaabane Djéraba. Bina-rep event frames: A simple and effective representation for event-based cameras. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3998–4002. IEEE, 2022. 4

[4] Sami Barchid, José Mennesson, Jason Eshraghian, Chaabane Djéraba, and Mohammed Bennamoun. Spiking neural networks for frame-based and event-based single object localization. *arXiv preprint arXiv:2206.06506*, 2022. 4

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 2, 3, 4, 6

[6] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 29:9084–9098, 2020. 5, 6, 8

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 2

[8] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[10] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R Risbud. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021. 1

[11] Wei Fang, Yanqi Chen, Jianhao Ding, Ding Chen, Zhaofei Yu, Huihui Zhou, Yonghong Tian, and other contributors. Spikingjelly. https://github.com/fangwei123456/spikingjelly, 2020. Accessed: 2022-02-01. 6

[12] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

[13] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1, 3

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[15] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007. 6

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2

[17] Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. Eventdrop: Data augmentation for event-based learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 700–707. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track. 3, 4, 7

[18] Matthew Gwilliam and Abhinav Shrivastava. Beyond supervised vs. unsupervised: Representative benchmarking and analysis of image representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9642–9652, 2022. 2, 5, 6, 8

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[20] 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. 3

[21] 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. 2

[22] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2146–2156, 2021. 2, 5

[23] Youngeun Kim, Joshua Chough, and Priyadarshini Panda. Beyond classification: Directly training spiking neural networks for semantic segmentation. *arXiv preprint arXiv:2110.07742*, 2021. 1

[24] Simon Klenk, David Bonello, Lukas Koestler, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. *arXiv preprint arXiv:2212.10368*, 2022. 2, 5, 8

[25] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network represen-

tations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 2

[26] Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations., July 2021. Documentation available under https://tonic.readthedocs.io. 3, 4

[27] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 631–649. Springer, 2022. 2, 3, 4

[28] Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *IJCAI*, pages 1743–1749, 2021. 2, 5, 6, 8

[29] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[30] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 4

[31] Fariborz Baghaei Naeini, Sanket Kachole, Rajkumar Muthusamy, Dimitrios Makris, and Yahya Zweiri. Event augmentation for contact force measurements. *IEEE Access*, 10:123651–123660, 2022. 3

[32] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. 1, 4

[33] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015. 2, 5

[34] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2021. 2

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[36] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 2

[37] Guobin Shen, Dongcheng Zhao, and Yi Zeng. Eventmix: An efficient augmentation strategy for event-based data. *arXiv preprint arXiv:2205.12054*, 2022. 3

[38] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018. 5, 6

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *CoRR*, abs/1711.11248, 2017. 3

[40] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021. 6

[41] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 6, 8

[42] Yan Yang, Liyuan Pan, and Liu Liu. Event camera dat a pre-training. *arXiv preprint arXiv:2301.01928*, 2023. 2, 5, 8

[43] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 2, 3

[44] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. 2