Neuromorphic Data Augmentation for Training Spiking Neural Networks

Yuhang Li, Youngeun Kim, Hyoungseob Park, Tarmar Geller, Priyadarshini Panda
Yale University, USA

➢ Observation
(1)
(2) Normalized D_{KL} is fairly consistent across training epochs

Introduction

➢ Event-based Data

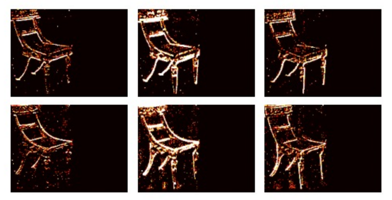
- Streaming data with only 0 or 1 in its form.
- Each pixel is operated independently and asynchronously, reporting new brightness when it changes, and staying silent otherwise.

RGB Data

3-D: width, higher, and color channel

Static, non-streaming

Event-based Data



4-D: width, higher, time, and polarity

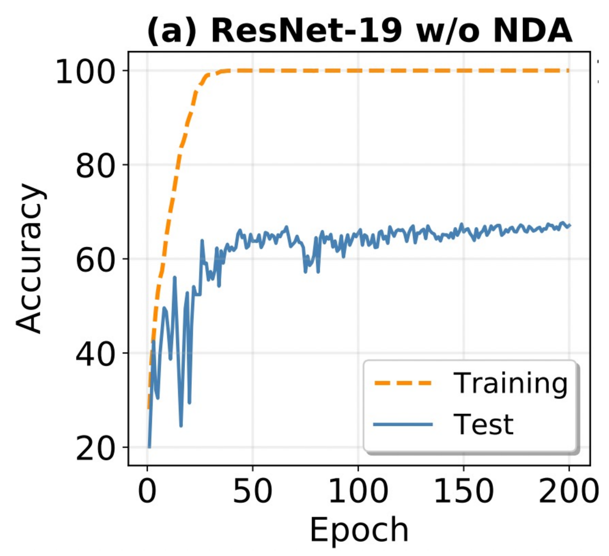
Dynamic, streaming

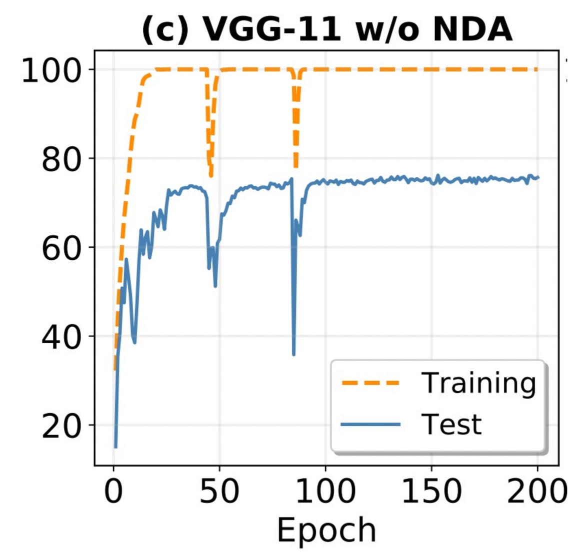
➢ Spiking Neural Networks

- SNNs inherently can model streaming data with time steps.
- The spiking neurons constantly take input and output spikes through time
- Moreover, the spike are binary, like events.
- However, most existing works focus on the model improvement, they ignores the dataset issues.

➢ Problems of Event-based Datasets

- Generating Event-based data is expensive.
- For example, DVS CIFAR-10 contains 10k examples.
- Therefore, SNNs easily get over-parameterized on Event-based dataset
- ResNet-19 / VGG 11 on DVS-CIFAR10 training curve

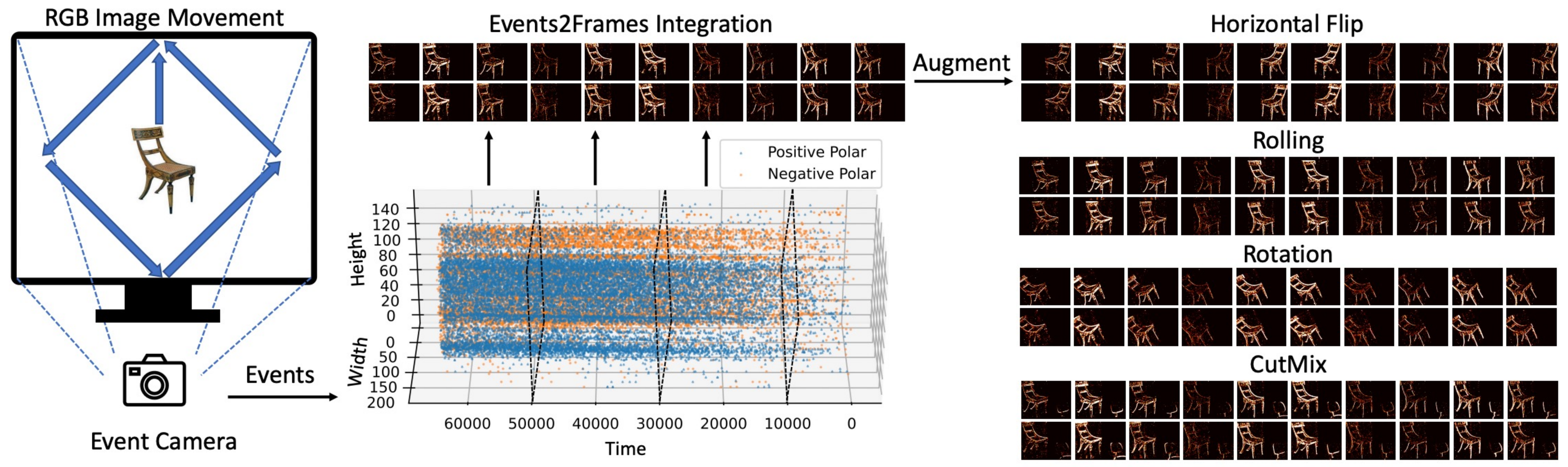




Neuromorphic Data Augmentation

➢ Since generating Event-based data is expensive, requiring special camera and human expert.

Can we directly augment the off-the-shelf event-based data to avoid converting augmented RGB data to events?



➢ NDA study the transform-invariant property:

$$f(H_\alpha[\log V(t)-\log V(t-\Delta t)]) \approx H_\alpha[\log f(V(t))-\log f(V(t-\Delta t))],$$

Aug.	Combination	Input-output	Pros	Cons
f_P	$g \circ f_P(\boldsymbol{x})$	$\mathbb{C}^{3\times w\times h} \to \mathbb{B}^{t\times p\times w\times h}$	i. Effective augmentation	i. Impractical to record huge amount of DVS data
f_P	$f_P \circ g(\boldsymbol{x})$	$\mathbb{C}^{3\times w\times h} \to \mathbb{C}^{t\times p\times w\times h}$	i. Practical	i. Not effective, ii. Creates continuous data
f_G	$g \circ f_G(\boldsymbol{x})$	$\mathbb{C}^{3\times w\times h} \to \mathbb{B}^{t\times p\times w\times h}$	i. Effective augmentation	i. Impractical to record huge amount of DVS data
f_G	$f_G \circ g(\boldsymbol{x})$	$\mathbb{C}^{3\times w\times h} \to \mathbb{B}^{t\times p\times w\times h}$	i. Practical and effective, ii. Approximates $g \circ f_G(\boldsymbol{x})$	None

➢ NDA selects geometrical augmentations:

- Flip, Roll, Rotate, Cutout, Shear, CutMix

Experiments

➢ Comparison to Existing Literature

- NDA improves 5—15% accuracy.

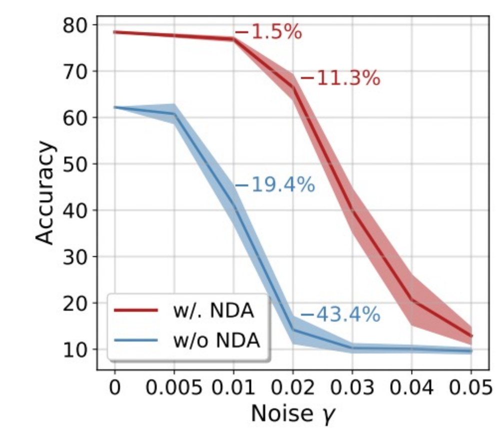
Table 4: Accuracy comparison with different methods on CIFAR10-DVS, N-Caltech 101, N-Cars, we use tdBN in our model. Acc. is referred as the top-1 accuracy.

Method	Model	CIFAR10-DVS		N Caltech-101		N-Cars	
		T Step	Acc.	T Step	Acc.	T Step	Acc.
HOTS [35]	N/A	N/A	27.1	N/A	21.0	10	54.0
Gabor-SNN [57]	2-layer CNN	N/A	28.4	N/A	28.4	-	-
HATS [57]	N/A	N/A	52.4	N/A	64.2	10	81.0
DART [48]	N/A	N/A	65.8	N/A	66.8	-	-
CarSNN [60]	4-layer CNN	-	-	-	-	10	77.0
CarSNN [60]	4-layer CNN[2]	-	-	-	-	10	86.0
BNTT [32]	6-layer CNN	20	63.2	-	-	-	-
Rollout [34]	VGG-16	48	66.5	-	-	-	-
SALT [33]	VGG11	20	67.1	20	55.0	-	-
LIAF-Net [65]	VGG-like	10	70.4	-	-	-	-
tdBN [69]	ResNet-19[1]	10	67.8	-	-	-	-
PLIF [17]	VGG-11[2]	20	74.8	-	-	-	-
tdBN (w/o NDA)[3]	ResNet-19[1]	10	67.9	10	62.8	10	82.4
tdBN (w/. NDA)	ResNet-19[1]	10	**78.0**	10	**78.6**	10	**87.2**
tdBN (w/o NDA)[3]	VGG-11	10	76.2	10	67.2	10	84.4
tdBN (w/. NDA)	VGG-11	10	**79.6**	10	**78.2**	10	**90.1**
tdBN (w/o NDA)[3]	VGG-11[2]	10	76.3	10	72.9	10	87.4
tdBN (w/. NDA)	VGG-11[2]	10	**81.7**	10	**83.7**	10	**91.9**

[1] Quadrupled channel number, [2] 128×128 resolution, [3] Our implementation.

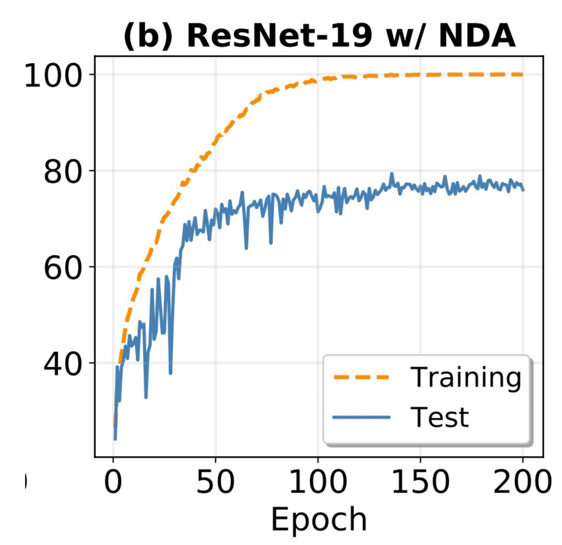
➢ Model sharpness comparison

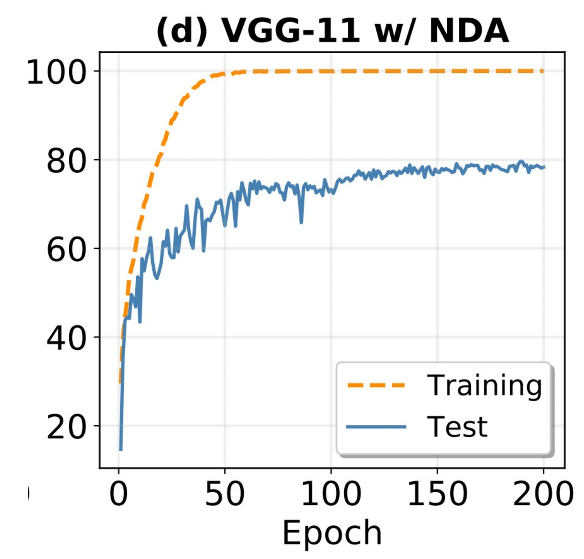
- NDA improves SNN's robustness to noise imposed on the weight parameters.
- NDA reduce Hessian spectrum.



Epoch	w/o NDA			w/. NDA		
	λ_1	λ_5	Tr	λ_1	λ_5	Tr
100	910.7	433.4	6277	424.3	73.87	1335
200	3375	1416	21342	516.4	155.3	1868
300	3404	1686	20501	639.7	187.5	2323

➢ NDA slightly increases the fire rate (10%).





Code is aviliable!
https://github.com/Intelligent-Computing-Lab-Yale/NDA-SNN