Supplementary Material: Deep Residual Learning in Spiking Neural Networks

Wei Fang[1,2], Zhaofei Yu[1,2*], Yanqi Chen[1,2],

Tiejun Huang[1,2], Timothée Masquelier[3], Yonghong Tian[1,2*]

[1]Department of Computer Science and Technology, Peking University
[2]Peng Cheng Laboratory, Shenzhen 518055, China
[3]Centre de Recherche Cerveau et Cognition, UMR5549 CNRS - Univ. Toulouse 3 , Toulouse, France

A Appendix

A.1 Hyper-Parameters

For all datasets, the surrogate gradient function is $\sigma(x) = \frac{1}{\pi}\arctan(\frac{\pi}{2}\alpha x) + \frac{1}{2}$, thus $\sigma'(x) = \frac{\alpha}{2(1+(\frac{\pi}{2}\alpha x)^2)}$, where α is the slope parameter. We set $\alpha = 2$, $V_{reset} = 0$ and $V_{th} = 1$ for all neurons. The optimizer is SGD with momentum 0.9. As recommended by [9], we detach $S[t]$ in the neuronal reset Eq. (3) in the backward computational graph to improve performance. We use the mixed precision training [7], which will accelerate training and decrease memory consumption, but may cause slightly lower accuracy than using full precision training. The hyper-parameters of the SNNs for different datasets are shown in Tab. S1. Tab. S2 shows the learning rates of the SNNs with different element-wise functions for DVS Gesture. The data pre-processing methods for three datasets are as following:

ImageNet The data augmentation methods used in [4] are also applied in our experiments. A 224×224 crop is randomly sampled from an image or its horizontal flip with data normalization for train samples. A 224×224 resize and central crop with data normalization is applied for test samples.

DVS128 Gesture We use the same AER data pre-processing method as [2], and utilize *random temporal delete* to relieve overfitting, which is illustrated in Sec. A.2.

CIFAR10-DVS We use the same AER data pre-processing method as DVS128 Gesture. We do not use *random temporal delete* because CIFAR10-DVS is obtained by static images.

Dataset	**Learning Rate Scheduler**	**Epoch**	lr	**Batch Size**	T	n_{gpu}
ImageNet	Cosine Annealing [6], $T_{max} = 320$	320	0.1	32	4	8
DVS Gesture	Step, $T_{step} = 64.\gamma = 0.1$	192	0.1	16	16	1
CIFAR10-DVS	Cosine Annealing, $T_{max} = 64$	64	0.01	16	4, 8, 16	1

Table S1: Hyper-parameters of the SNNs for three datasets.

A.2 Random Temporal Delete

To reduce overfitting, we propose a simple data augmentation method called *random temporal delete* for sequential data. Denote the sequence length as T, we randomly delete $T - T_{train}$ slices in the

*Corresponding author

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

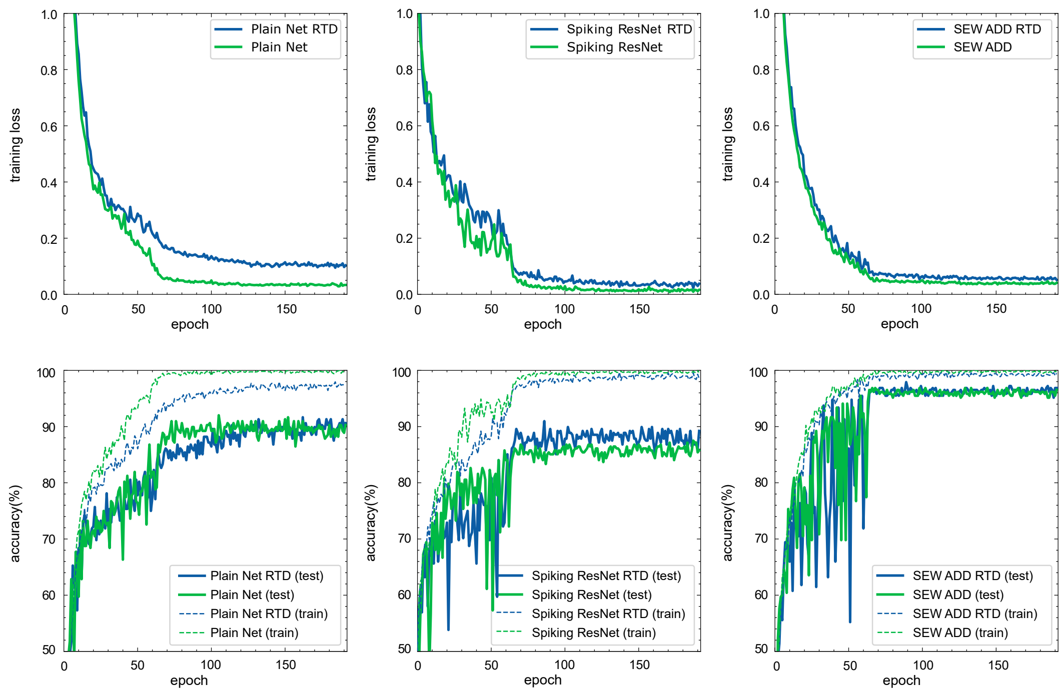


Figure S1: Comparison of training loss and training/test accuracy with/without random temporal delete (RTD).

Network	Element-Wise Function g	Learning Rate
SEW ResNet	ADD	0.001
SEW ResNet	AND	0.03
SEW ResNet	IAND	0.063
Spiking ResNet	-	0.1
Plain Net	-	0.005

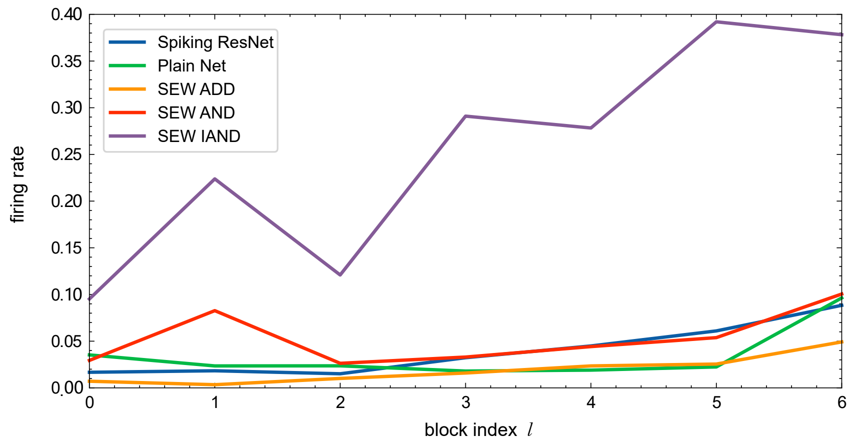
Table S2: Learning rates of the SNNs for DVS Gesture.

origin sequence and use T_{train} slices during training. During inference we use the whole sequence, that is, $T_{test} = T$. We set $T_{train} = 12, T = 16$ in all experiments on DVS Gesture.

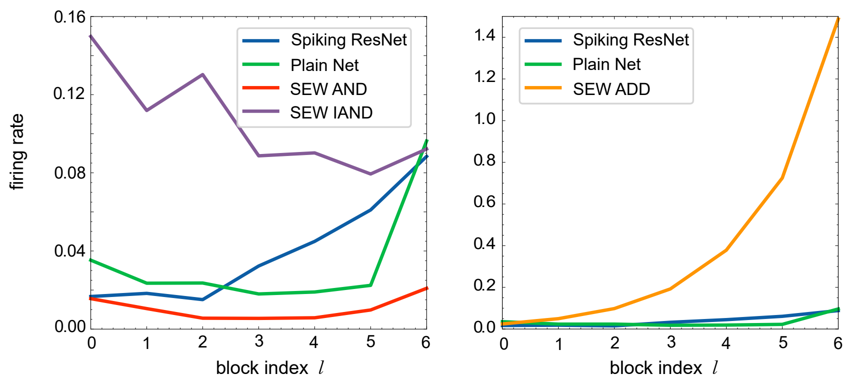
Fig. S1 compares the training loss and training/test accuracy of Plain Net, Spiking ResNet, and SEW ResNet with or without *random temporal delete* (RTD). Here the element-wise function g is *ADD*. It can be found that the network with RTD has higher training loss and lower training accuracy than the network without RTD, because RTD can increase the difficulty of training. The test accuracy of the network with RTD is higher than that without RTD, showing that RTD will reduce overfitting. The results on the three networks are consistent, indicating that RTD is a general sequential data augmentation method.

A.3 Firing rates on DVS Gesture

Fig. S2(a) shows the firing rates of A^l in each block from 7B-Net for DVS Gesture. Note that if g is *AND*, the SEW block gets closer to identity mapping when the firing rate approaches 1, while for other g, the SEW block becomes identity mapping when the firing rate approaches 0. When all SEW blocks become identity mapping, the 7B-Net will become *c32k3s1-BN-PLIF-{MPk2s2}*7-FC11*, which is a too simple network to cause underfitting. Thus, the SEW blocks in 7B-Net are not necessary to be identity mapping. Fig. S2(b) shows the firing rates of each block's output O^l. The firing rates do not strictly decrease with block index increases as blocks are connected by max pooling, which squeezes sparse spikes and increases the firing rate. It can be found that the blocks in SEW AND network have



(a) Firing rates of A^l in each block on DVS Gesture Gesture



(b) Firing rates of the output O^l in each block on DVS Gesture Gesture

Figure S2: Firing rates of the last SN and the output O^l in each block of 7B-Net on DVS Gesture.

the lowest firing rates. The blocks in SEW IAND network have higher firing rates than those of SEW AND network, and the SEW IAND network has much higher accuracy than the SEW AND network (95.49% v.s. 70.49%), indicating that using *IAND* to replace *AND* can relieve the silence problem discussed in Sec.4.1.

A.4 Gradients in Spiking ResNet with Firing Rates

The gradients of SNNs are affected by firing rates, which is the reason why we analyze the firing rates before gradients in Sec.4.1. Consider a spiking ResNet with k sequential blocks to transmit $S^l[t]$, and the identity mapping condition is met, e.g., the spiking neurons are the IF neurons with $0 < V_{th} \leq 1$, then we have $S^l[t] = S^{l+1}[t] = ... = S^{l+k-1}[t] = O^{l+k-1}[t]$. We get

$$\frac{\partial O_j^l[t]}{\partial S_j^l[t]} = \frac{\partial \mathrm{SN}(S_j^l[t])}{\partial S_j^l[t]} = \Theta'(S_j^l[t] - V_{th}) \tag{S1}$$

$$\frac{\partial L}{\partial S_j^l[t]} = \frac{\partial L}{\partial O_j^l[t]} \Theta'(S_j^l[t] - V_{th}). \tag{S2}$$

Then the gradients between two adjacent blocks are

$$\frac{\partial L}{\partial O^{l+i}} = \frac{\partial L}{\partial O^{l+i+1}} \Theta'(S^{l+i+1} - V_{th}). \tag{S3}$$

Surrogate function	SEW ResNet	Spiking ResNet
ArcTan	0.8263	0.7733
Rectangular	0.8256	0.6601
Constant 1	0.1256	0.1

Table S3: Test accuracy of SEW ADD ResNet and Spiking ResNet on CIFAR-10 with different surrogate functions.

Denote the number of neurons as N, the firing rate of S^l as $\Phi = \frac{\sum_{j=0}^{N-1}\sum_{t=0}^{T-1} S_j^l[t]}{NT}$, then

$$\left\|\frac{\partial L}{\partial S^l}\right\| = \left\|\frac{\partial L}{\partial O^{l+k-1}}\right\| \cdot \left\|\prod_{i=0}^{k-1} \Theta'(S^{l+i} - V_{th})\right\|, \tag{S4}$$

where

$$\begin{aligned}\left\|\prod_{i=0}^{k-1} \Theta'(S^{l+i} - V_{th})\right\| &= \sqrt{NT\Phi(\Theta'(1-V_{th}))^k + NT(1-\Phi)(\Theta'(0-V_{th}))^k} \\ &\to \begin{cases} \sqrt{NT}, & \Theta'(1-V_{th}) = 1, \Theta'(0-V_{th}) = 1 \\ \sqrt{NT\Phi}, & \Theta'(1-V_{th}) = 1, \Theta'(0-V_{th}) < 1 \\ \sqrt{NT(1-\Phi)}, & \Theta'(1-V_{th}) < 1, \Theta'(0-V_{th}) = 1 \\ 0, & \Theta'(1-V_{th}) < 1, \Theta'(0-V_{th}) < 1 \\ +\infty, & \Theta'(1-V_{th}) > 1 \ or \ \Theta'(0-V_{th}) > 1. \end{cases}\end{aligned}$$

A.5 0/1 Gradients Experiments

As the analysis in Sec.3.2 shows, the vanishing/exploding gradient problems are easy to happen in Spiking ResNet because of accumulative multiplication. A potential solution is to set $\Theta'(0 - V_{th}) = \Theta'(1 - V_{th}) = 1$. Specifically, we have trained the Spiking ResNet on ImageNet by setting $V_{th} = 0.5$ and $\sigma'(x) = \frac{1+\frac{\pi^2}{4}}{1+(\pi x)^2}$ in the last SN of each block to make sure that $\Theta'(0 - V_{th}) = \Theta'(1 - V_{th}) = 1$. However, this network will not converge, which may be caused by that SNNs are sensitive to surrogate functions.

[10] uses the Rectangular surrogate function $\sigma'(x) = \frac{1}{a}\mathrm{sign}(|x| < \frac{a}{2})$. If we set $a = 1$, then $\sigma'(x) \in \{0, 1\}$. According to Eq.(8), using this surrogate function can avoid the gradient exploding/vanishing problems in Spiking ResNet. We compare different surrogate functions, including Rectangular ($\sigma'(x) = \mathrm{sign}(|x| < \frac{1}{2})$), ArcTan ($\sigma'(x) = \frac{1}{1+(\pi x)^2}$) and Constant 1 ($\sigma'(x) \equiv 1$), in the SNNs on CIFAR-10. Note that we aim to evaluate 0/1 gradients, rather than achieve SOTA accuracy. Hence, we use a lightweight network, whose structure is *c32k3s1-BN-IF-{{SEW Block(c32k3s1)}*2-MPk2s2}*5-FC10*. We use *ADD* as g in SEW blocks. We also compare with Spiking ResNet by replacing SEW blocks with basic blocks. The results are shown in Tab.S3. The learning rates for each surrogate function are fine-tuned.

Tab.S3 indicates that the choice of surrogate function has a considerable influence on the SNN's performance. Although Rectangular and Constant 1 can avoid the gradient exploding/vanishing problems in Eq.(8), they still cause lower accuracy or even make the optimization not converges. Tab.S3 also shows that the SEW ResNet is more robust to the surrogate gradient as it always has higher accuracy than the Spiking ResNet with the same surrogate function.

A.6 Reproducibility

All experiments are implemented with SpikingJelly [1], which is an open-source deep learning framework for SNNs based on PyTorch [8]. Source codes are available at `https://github.com/fangwei123456/Spike-Element-Wise-ResNet`. To maximize reproducibility, we use identical seeds in all codes.

References

[1] Wei Fang, Yanqi Chen, Jianhao Ding, Ding Chen, Zhaofei Yu, Huihui Zhou, Yonghong Tian, and other contributors. Spikingjelly. `https://github.com/fangwei123456/spikingjelly`.

[2] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2661–2671, 2021.

[3] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

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

[5] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017.

[6] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[7] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *International Conference on Learning Representations (ICLR)*, 2018.

[8] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, 2019.

[9] Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021.

[10] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11062–11070, 2021.