

# REVISITING BATCH NORMALIZATION FOR TRAINING LOW-LATENCY DEEP SPIKING NEURAL NETWORKS FROM SCRATCH

**Anonymous authors**

Paper under double-blind review

## A APPENDIX: DVS-CIFAR10 DATASET

On DVS-CIFAR10, following [Wu et al. \(2019\)](#), we downsample the size of the  $128 \times 128$  images to  $42 \times 42$ . Also, we divide the total number of time-steps available from the original time-frame data into 20 intervals and accumulate the spikes within each interval. We use a similar architecture as previous work ([Wu et al. 2019](#)), which consists of a 5-layered feature extractor and a classifier. The detailed architecture is shown in Fig. 1 in this appendix.

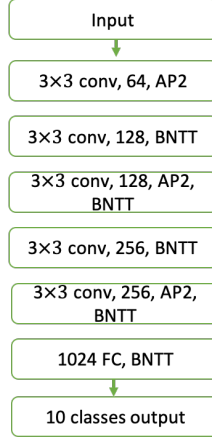


Figure 1: Illustration of network structures for DVS dataset. Here, AP denotes average pooling, FC denotes fully connected configuration.

## B APPENDIX: ENERGY CALCULATION

In this appendix section, we provide the details of energy calculation discussed in Section 4.3 in the main paper. The total computational cost is proportional to the total number of floating point operations (FLOPS). This is approximately the same as the number of Matrix-Vector Multiplication (MVM) operations. For layer  $l$  in ANNs, we can calculate FLOPS as:

$$FLOPS_{ANN}(l) = \begin{cases} k^2 \times O^2 \times C_{in} \times C_{out}, & \text{if } l \text{ is Convolution Layer,} \\ C_{in} \times C_{out}, & \text{if } l \text{ is Linear Layer.} \end{cases} \quad (1)$$

Here,  $k$  is kernel size.  $O$  is output feature map size.  $C_{in}$  and  $C_{out}$  are input and output channels, respectively. For SNNs, we first define spiking rate  $R_s(l)$  at layer  $l$  which is the average firing rate per a neuron.

$$R_s(l) = \frac{\# \text{spikes of layer } l \text{ over all timesteps}}{\# \text{neurons of layer } l}. \quad (2)$$

Since neurons in SNNs consume energy whenever the neurons are activated, we multiply the spiking rate  $R_s(l)$  with FLOPS.

$$FLOPS_{SNN}(l) = FLOPS_{ANN}(l) \times R_s(l). \quad (3)$$

Finally, total inference energy of ANNs ( $E_{ANN}$ ) and SNNs ( $E_{SNN}$ ) across all layer can be obtained.

$$E_{ANN} = \sum_l FLOPS_{ANN}(l) \times E_{MAC}. \quad (4)$$

$$E_{SNN} = \sum_l FLOPS_{SNN}(l) \times E_{AC}. \quad (5)$$

The  $E_{AC}$ ,  $E_{MAC}$  values are calculated using a standard 45 nm CMOS process (Horowitz 2014) as shown in Table 1.

Table 1: Energy table for 45nm CMOS process.

Operation	Energy(pJ)
32bit FP MULT ( $E_{MULT}$ )	3.7
32bit FP ADD ( $E_{ADD}$ )	0.9
32bit FP MAC ( $E_{MAC}$ )	4.6 (= $E_{MULT} + E_{ADD}$ )
32bit FP AC ( $E_{AC}$ )	0.9

## REFERENCES

- Mark Horowitz. 1.1 computing’s energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. IEEE, 2014.
- Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1311–1318, 2019.