Brain-Inspired Spiking Neural Networks for Energy-Efficient Object Detection

Supplementary Material

1. Interpretability Analysis of Gradient Vanishing/Explosion Problems

We theoretically analyzed the feasibility of implementing deep training for MSD and explained why it demonstrates exceptional performance. Based on Gradient Norm Equality (GNE) theory [1], MSD can effectively prevent gradient vanishing and explosion.

1.1. Lemma 1

A neural network can be considered as consisting of multiple blocks, where the Jacobian matrix of the j-th block is denoted as J_j. If $\forall j$; $\phi(J_j, J_j^{\top}) \approx 1$ and $\varphi(J_j, J_j^{\top}) \approx 0$, the network achieves "Block Dynamical Isometry", which could prevent gradient vanishing or explosion.

Wherein, J_j denotes the Jacobian matrix of the block j, where j is the index of the corresponding block. φ represents $\varphi(A^2) - \varphi^2(A)$. The GNE theory ensures that the network's gradients neither vanish to 0 nor explode to ∞, as each block maintains $\phi(J_j, J_j^{\top}) \approx 1$ ensures the avoidance of anomalous situations. In most cases[2, 9], $\phi(J_j, J_j^{\top}) \approx 1$ is sufficient to prevent gradient vanishing or explosion[1].

Theorem 1. General Linear Transform $f(x)$ is a transformation whose Jacobian matrix is J. It is referred to as a General Linear Transformation if it satisfies the following conditions:

$$E\left[\frac{\|f(x)\|_2^2}{len(f(x))}\right] = \phi(J, J^{\top}) E\left[\frac{\|x\|_2^2}{len(x)}\right] \tag{1}$$

Based on random matrix theory and mean field theory[4], the data flow propagated through the network can be regarded as a random variable. Wherein, x is treated as a random variable, $E\left[\frac{\|x\|_2^2}{len(x)}\right]$ denotes the 2-th moment of input element. This definition is useful for gradient analysis because, once the output of the EMS block is normalized by the BN layer, the second moment $E\left[\frac{\|f(x)\|_2^2}{len(f(x))}\right]$ is represented as α_2.

1.2. Lemma 2: Multiplication

As theorem 4.1 in [1], given $J = \prod_{j=L}^{1} J_j$, where $J_j = \mathbb{R}^{m_j \times m_{j-1}}$ is a series of independent random matrices. If $(J = \prod_{j=L}^{1} J_j)(J = \prod_{j=L}^{1} J_j)^{\top}$ is at least unitarily invariant at the 1_{st} moment, then we could obtain:

$$\phi\left((\prod_{j=L}^{1} J_j)(\prod_{j=L}^{1} J_j)^{\top} \right) = \prod_{j=L}^{1} \phi(J_j, J_j^{\top}) \tag{2}$$

1.3. Lemma 3: Addition

As theorem 4.2 in [1], given $J = \prod_{j=L}^{1} J_j$, where $J_j = \mathbb{R}^{m_j \times m_{j-1}}$ is a series of independent random matrices. If at most one matrix in J_j is not a centered matrix, then we have

$$\phi(J, J^{\top}) = \sum_j \phi(J_j, J_j^{\top}) \tag{3}$$

The principles of multiplication and addition provide a method for analyzing both serial and parallel networks.

Discussion on General Linear Transformations. The Jacobian matrix of pooling can be represented as a matrix J, where each element $[J]_{ik} \in \{0, 1\}$. For elements that are not selected, $[J]_{ik} = 0$, and for selected elements, $[J]_{ik} = 1$. Therefore, the pooling layer can be viewed as a general linear transformation. Similarly, the UpSampling layer in the MSDF is also a generalized linear transformation. Concatenation is also a general linear transformation, as the function $f(x) = [x, \tilde{f}(x)]$ can be expressed as $f(x) = [I \;\; \tilde{J}]x$, where $\tilde{f}$ is a general linear transformation. Batch normalization and convolution layers have been discussed in [1], so we only need to assume that the Leaky Integrate-and-Fire (LIF) layer satisfies a general linear transformation, which has been proven in [9]. Since MSD consists of a series of SCNs and ONNBs, we can analyze these blocks individually and multiply their effects accordingly.

Proposition 1. For SCN and ONNB, the Jacobian matrix of the block can be expressed as:

$$\phi(J_j, J_j^{\top}) = \frac{2}{\alpha_2^{j-1}} \tag{4}$$

Proposition 2. For MSD, $\phi(J_j, J_j^{\top}) \approx 1$ can be satisfied by control the 2-th moment of the input.

Insights into the Proposition. According to [1], $\varphi(JJ^{\top}) \approx 1$ represents the formal expression of GNE, which ensures that gradients neither vanish nor explode. However, avoiding the exponential growth or decay described in [1] is sufficient to address gradient-related issues.

In our backbone network, the SCN is the only factor causing exponential gradient growth, which is mitigated by the interspersed ONNB. Even if the initialized batch normalization does not strictly satisfy $\varphi(JJ^{\top}) \approx 1$, the gradient of each block does not increase as the network deepens. In summary, our ONNB enhance the network's performance structurally.

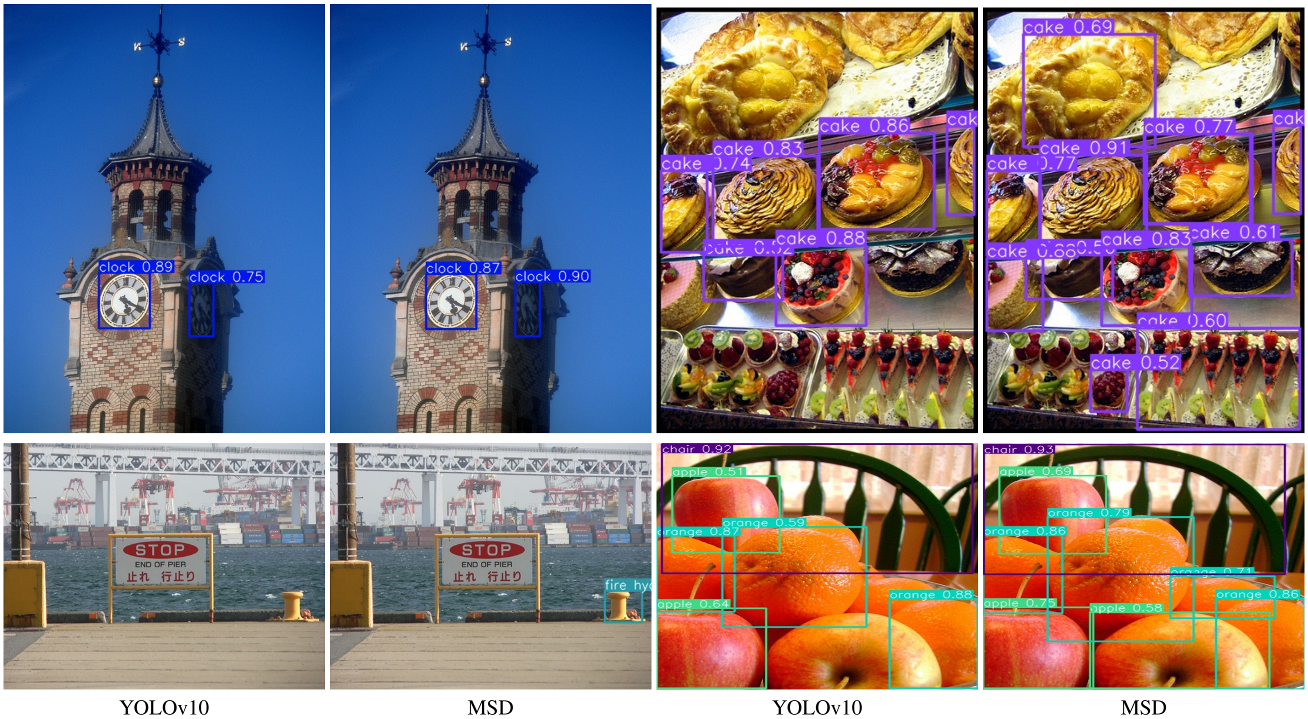

Figure 1. Object detection results on the COCO 2017 dataset. The figure compares the detection performance of YOLOv10 and MSD in the same scenario.

1.4. Gradient Norm Equality Proof

As the primary component of ONNB, spiking neuron (SN) contains a residual path and a shortcut path, which Jacobian matrices are denoted as J_{res} and J_{sc}. n denotes the number of layers. As shown in Fig. 3, the General Linear Transform of the two paths is expressed as:

$$\alpha_2^{l,res} = \phi(J_{res}, J_{res}^\top)\alpha_2^{l-1}, \tag{5}$$

$$\alpha_2^{l,sc} = \phi(J_{sc}, J_{sc}^\top)\alpha_2^{l-1}, \tag{6}$$

wherein, α_2^{l-1} is the 2^{th} moment of the input data from $(l-1)^{th}$ block. The output of the BN layer has a variance of 1 and a mean of 0, $\alpha_2^{l,res} = \alpha_2^{l,sc} = 1$. Therefore,

$$\phi(J_{res}, J_{res}^\top) = \frac{1}{\alpha_2^{l-1}}, \tag{7}$$

$$\phi(J_{sc}, J_{sc}^\top) = \frac{1}{\alpha_2^{l-1}}. \tag{8}$$

According to the discussion about concatenation [1] in supplementary, we derive

$$\phi(J_j, J_j^\top) = \frac{c_{j-1}}{c_j} + \frac{\delta_j}{c_j}\phi(H_j, H_j^\top), \tag{9}$$

wherein, J_j denotes Jacobian matrix of shortcut path without maxpooling layer. H_j denote the Jacobian matrix of the SCN. C_{j-1} and C_j represent the input and output channels of the concatenation operation. $\delta_j = c_j - c_{j-1}$. By adding the maxpooling layer, shortcut path can be expressed as:

$$\begin{aligned}\phi(J_{sc}, J_{sc}^\top) &= \frac{\alpha_2^{maxpool}}{\alpha_2^{l-1}}\left(\frac{c_{j-1}}{c_j} + \frac{\delta_j}{c_j}\phi(H_j, H_j^\top)\right) \\ &= \frac{1}{\alpha_2^{l-1}}\left(\frac{\alpha_2^{maxpool}c_{j-1}}{c_j} + \frac{\alpha_2^{maxpool}\delta_j}{c_j}\left(\frac{\alpha_2^{bn}}{\alpha_2^{maxpool}}\right)\right).\end{aligned} \tag{10}$$

Since the 2^{th} moment α_2^{l-1} is strictly controlled by the BN layers of former block and $\alpha_2^{maxpool}$ is fixed. Proper initializing of BN layers, with $\alpha_2^{bn} = \frac{2c_j - \alpha_2^{maxpool}c_{j-1}}{\delta_j}$, ensures that $\phi(J_{sc}, J_{sc}^\top) = \frac{1}{\alpha_2^{l-1}}$ holds.

$$\phi(J_{SN}, J_{SN}^\top) = \phi(J_{res}, J_{res}^\top) + \phi(J_{sc}, J_{sc}^\top) = \frac{2}{\alpha_2^{l-1}}. \tag{11}$$

For SN, $\phi(J, J^\top) \approx 1$ could be satisfied by control the 2^{th} moment of the input.

Resblock have already been discussed in [1]. Using general linear transform and addition principle, we derive

$$\alpha_2^{l-1}\phi(J, J^\top) = \alpha_2^l = \alpha_2^{l-1} + 1. \tag{12}$$

α_2^{l-1} originates from the preceding layer and is fixed at 2. Therefore, we obtain $\phi(J_{ONNB}, J_{ONNB}^\top) = \frac{3}{2}$. By using multiplication principle, the whole blocks have the property:

$$\phi(J, J^\top) = \frac{3}{\alpha_2^3}, \tag{13}$$

where α_2^0 is the 2^{th} moment of BN output in encoding layer. After initialized the BN in encoding layer, α_2^0 s controlled to

3 and $\phi(J, J^{\top}) \approx 1$ holds. Therefore, for MSD, $\phi(J, J^{\top}) \approx 1$ holds, which achieves "Block Dynamical Isometry" and prevents gradient vanishing or explosion.

2. Additional Experiments

We conducted additional experiments to compare the performance of our proposed method with several state-of-the-art ANN object detectors on MS COCO [3] benchmarks.

Table 1. Comparison of baseline object detectors on MS COCO

Method	Params(M)	Size	mAP@0.5	@mAP@0.5:0.95
PVT[7]	32.9	640	59.2	36.7
DETR[10]	41.0	640	62.4	42.0
YOLOV5-S	7.2	640	56.8	537.4
YOLOv7-tiny[6]	6.2	640	56.7	-
PPYOLOE-S[8]	7.9	640	60.5	46.6
YOLOv10-s[5]	8.05	640	59.1	42.3
MSD	7.8	640	62.0	45.3

We compared MSD with the baselines of recent high-performance object detectors. The results show that MSD achieves performance comparable to ANNs while maintaining significantly lower parameter counts and energy consumption.

Additionnaly, We compared the detection performance of MSD with YOLOv10 in the same scenario. MSD successfully detected targets that were missed by YOLOv10 and showed higher accuracy for the same targets, demonstrating the effectiveness of the proposed method in handling challenging scenarios.

References

[1] Zhaodong Chen, Lei Deng, Bangyan Wang, Guoqi Li, and Yuan Xie. A comprehensive and modularized statistical framework for gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):13–31, 2022. 1

[2] Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, Cham, 2014. Springer International Publishing. 3

[4] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016. 1

[5] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. 3

[6] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 3

[7] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 3

[8] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, et al. Pp-yoloe: An evolved version of yolo. *arXiv e-prints*, pages arXiv–2203, 2022. 3

[9] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 11062–11070, 2021. 1

[10] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 3