Advancing Deep Residual Learning by Solving the Crux of Degradation in Spiking Neural Networks

Yifan Hu, Yujie Wu, Lei Deng, Guoqi Li,
Department of Precision Instrument, Tsinghua University

Abstract

Despite the rapid progress of neuromorphic computing, the inadequate depth and the resulting insufficient representation power of spiking neural networks (SNNs) severely restrict their application scope in practice. Residual learning and shortcuts have been evidenced as an important approach for training deep neural networks, but rarely did previous work assess their applicability to the characteristics of spike-based communication and spatiotemporal dynamics. This negligence leads to impeded information flow and the accompanying degradation problem. In this paper, we identify the crux and then propose a novel residual block for SNNs, which is able to significantly extend the depth of directly trained SNNs, e.g., up to 482 layers on CIFAR-10 and 104 layers on ImageNet, without observing any slight degradation problem. We validate the effectiveness of our methods on both frame-based and neuromorphic datasets, and our SRM-ResNet104 achieves a superior result of 76.02% accuracy on ImageNet, the first time in the domain of directly trained SNNs. The great energy efficiency is estimated and the resulting networks need on average only one spike per neuron for classifying an input sample. We believe our powerful and scalable modeling will provide a strong support for further exploration of SNNs.

Introduction

Spiking neural networks are a typical kind of brain-inspired models with their unique features of rich neuronal dynamics and diverse coding schemes. Different from traditional artificial neural networks (ANNs), SNNs are capable of encoding information in spatiotemporal dynamics and using asynchronous binary spiking activities for event-driven communication. Recent progress in neuromorphic computing has demonstrated their great energy efficiency (Pei et al. 2019; Mayr, Hoeppner, and Furber 2019; Akopyan et al. 2015; Davies et al. 2018). Theoretically, SNNs are at least as computationally powerful as ANNs and the universal approximation theorem also applies to SNNs (Maass 1997), so it is not surprising that SNNs have been reported in various domains, such as image classification (Wu et al. 2019), object detection (Kim et al. 2020) and tracking (Yang et al. 2019), speech recognition (Wu et al. 2020), light-flow estimation (Lee et al. 2020), and so forth. However, the status quo of deficiency in powerful SNN models seriously limits their capabilities for complex tasks in practice.

Conversion from a pre-trained model and surrogate gradient-based direct training are two mainstream approaches for obtaining an high-accuracy SNN model. The conversion method is free of the dilemma caused by the non-differentiable spiking activities and can implement the inference of deep SNNs with tens or even hundreds of layers, depending on the pre-trained ANN models adopted (Sengupta et al. 2019; Han, Srinivasan, and Roy 2020; Hu, Tang, and Pan 2018; Stockl and Maass 2021). Although comparable accuracy to the pre-trained models can be achieved, the method treats SNNs more as an alternative expression of ANNs, and hundreds of time steps are usually required to achieve a satisfying conversion loss. It thereby fails to achieve low-latency computation in practical applications. The latter method utilizes a surrogate gradient function to enable backpropagation (BP) through time in the direct training of SNNs. The networks learn to encode information effectively and consequently only need much fewer time steps than the conversion ones. In addition, they are inherently more suited for processing spatiotemporal data from the emerging AER-based (address-event-representation) sensors, where the pre-training of ANNs is usually not applicable.

Unfortunately, one prominent problem of the directly trained SNNs lies in the limited scale of models. Earlier works mainly focus on shallow structures and simple tasks, such as MNIST (Deng 2012). Inspired from the representation power of deep ANNs, recent works have gradually evolved from fully-connected networks to convolutional networks to more advanced ResNet, for example Zheng et al. (2021) report the successful training of spiking ResNet-34 on ImageNet. However, the plain transplantation of the canonical ResNet, a way almost all previous works have adopted, does not work appropriately for the training of SNNs and the symptom manifests itself in the degradation problem, causing an accuracy drop in both the training and testing as the network deepens. As a result, building a deeper SNN still appears to be arduous and fruitless. To this end, a specific residual block design is highly desirable for applying deep residual learning to SNNs.

We summarize our major contributions in this work as follows:
• We reveal the degradation problem when applying the canonical ResNet in SNN training, and then identify the crux that causes information loss between blocks.

• Motivated by the crux, we design several novel alternative residual blocks and assess their applicability for the direct training of deep SNNs. The resulting SRM-ResNet enables us to significantly extend the depth of directly trained SNNs, e.g., up to 482 layers on CIFAR-10 and 104 layers on ImageNet, without slight degradation problem.

• The advantageous feature of convolution with spiking inputs is reserved intentionally, and along with the binary and sparse spiking activities, great energy efficiency is validated in an operation-based estimation.

• We evaluate the effectiveness of our methods on various datasets and obtain accuracy results which are, to the best of our knowledge, much better than previous work in directly trained SNNs and competitive to the conversion methods via extremely fewer time steps.

The most straightforward way of training higher quality models is by increasing their size, especially given the availability of a large amount of labeled training data (Szegedy et al. 2015). In this work, we would like to see that deepening network structures could get rid of the degradation problem and always be a trustworthy way to achieve satisfying accuracy for the direct training of SNNs.

### Related Works

#### SNN Training Algorithms

There are two main routines to training an high-accuracy SNN model. The first is the conversion method. Its basic idea is that the continuous activation values in an ANN using ReLU can be approximated by the average firing rate of an SNN under the rate coding scheme. After training an ANN with certain structure restrictions and the BP algorithm, it is feasible to convert the pre-trained ANN into its spiking counterpart. Conversion-based SNNs maintain the smallest gap with ANNs in terms of accuracy and can be generalized to large-scale structures and datasets. Rueckauer et al. (2017) report the spiking versions of VGG-16 and GoogleNet. Hu, Tang, and Pan (2018) offer a compensation mechanism to reduce the discretization error, and obtain an accuracy of 72.75% on ImageNet with spiking ResNet-50. In the work of Stöckl and Maass (2021), it is allowed for spikes to carry time-varying multi-bit information so that more closely the original activation function can be emulated and large EfficientNet models can be trained. However, the conversion method also has its inherent defects. An accuracy gap will be caused by the constraints on ANN models and a long simulation duration with hundreds or thousands of time steps is required to complete an inference, which leads to extra delay and energy consumption.

The second routine is to utilize a surrogate gradient function, which constitutes a continuous relaxation of the non-smooth spiking during BP, to enable standard BP or BPTT for training an SNN from scratch. Direct training algorithms appear to be diverse in the selection of gradient functions (Wu et al. 2018) and coding schemes (Zhang et al. 2020, Zhou et al. 2021). Compared to the converted SNNS, the directly trained SNNs have a great advantage in the number of time steps, which can be particularly appealing for the implementation on power-efficient neuromorphic hardware. Moreover, they take neuronal dynamics into further consideration during training and the capability for processing spatiotemporal event-stream data is always regarded as a critical measurement. Unfortunately, the shallow structures of the current directly trained SNNs obstruct further exploration on large-scale datasets and in complex tasks. Zheng et al. (2021) propose TDBN to alleviate the problem and obtain directly trained ResNet-34/50, but there is a lack of discussion about how helpful this method can be for deepening the network and their results are not satisfying enough compared to the converted SNNs or ANNs. We also notice a contemporary work that looks into deep residual SNNs (Fang et al. 2021) by adding element-wise operations into two spiking branches within a res-block. Their method is similar but orthogonal to ours.

### Residual Learning

Theoretical and empirical evidences indicate that the depth of neural networks is crucial for their success, but the training becomes more difficult as the depth increases (Srivastava, Greff, and Schmidhuber 2015). Starting with ResNet (He et al. 2016a,b), the shortcut connection is introduced as an additional branch for an unimpeded flow of both information and its gradient. It is widely adopted in later works, such as DenseNet (Huang et al. 2017), Inception-ResNet (Szegedy et al. 2017), and Transformer (Vaswani et al. 2017). One explanation for its superiority is that ResNet works as an implicit ensemble of numerous shallower networks (Veit, Wilber, and Belongie 2016). In addition, via the perspective of dynamical mean field theory, ResNet is proved to achieve dynamical isometry in a universal way (Tarnowski et al. 2019). Obviously, residual learning has gone beyond the ResNet structure and becomes a key component in the network architecture design. It has been utilized in the converted SNNs but remains challenging in the directly trained SNNs. In our work, the drawbacks of adopting canonical ResNet in SNNs are revealed and an improved spiking residual block is proposed.

### Preliminaries and Motivation

#### Preliminaries of SNNs

The basic differences between an SNN and an ANN originate from their primary computing element, i.e., the neuron. In ANNs, a biological neuron is abstracted as an information aggregation unit with a nonlinear transformation. Meanwhile in SNNs, the neuronal dynamics of the membrane potential and the spiking communication scheme are more closely mimicked. Typically, there are several kinds of spiking neuron models, such as leaky integrate-and-fire (LIF) (Abbott 1999, Izhikevich 2003), and Hodgkin-Huxley (Hodgkin and Huxley 1952). At the level of large-scale neural networks, LIF neuron models are...
widely adopted due to the concise form and lower computational complexity. In this work, we select the iterative LIF model proposed by Wu et al. (2019), which is formulated as

\[ u_i = \tau_{\text{mem}} \cdot u_i^{t-1} + I_i, \]  
\[ I_i = \sum_{j=1}^{n} w_{ij} o_j^t, \]  
\[ o_i^t = g(u_i^t - V_{th}) = \begin{cases} 1, & u_i^t - V_{th} \geq 0 \\ 0, & u_i^t - V_{th} < 0 \end{cases}, \]  

where \( u_i^t \) is the membrane potential of the \( i^{th} \) neuron in a layer at time step \( t \), \( \tau_{\text{mem}} \) is a decay factor for leakage and the input \( I_i^t \) is the weighted sum of output spikes from the previous layer. \( g(\cdot) \) describes the firing activity controlled by the threshold \( V_{th} \) and \( u_i^t \) will be subsequently reset to \( V_{\text{reset}} \) once a spike fires.

The surrogate gradient is defined as

\[ \frac{\partial o_i^t}{\partial u_i^t} = \frac{1}{a} \text{sign}(|u_i^t - V_{th}| - \frac{a}{2}). \]  

The coefficient \( a \) is introduced to ensure that the integral of the function is 1. This surrogate function makes those neurons close to the firing threshold receive gradient information from the preceding layers. In this way, the standard BPTT can be carried out in SNNs along with the autograd framework of current deep learning libraries. The first layer in the network is trained as an implicit encoder, which is able to process both spiking and non-spiking inputs and always outputs spiking signals. This scheme is of great help to both task performance and efficient representation of input data within a limited number of time steps. At the end of our model, a fully-connected voting layer counts the number of spikes during the entire simulation and works as a classifier to make the final decision.

PlainNet, Transplantation of ResNet, and Degradation

A universal structure in ANNs is a stacking of \{Conv-BN-Nonlinearity\} unit, which follows the philosophy of VGG network. As shown in Fig. 1, the corresponding construction of SNNs can be regarded as a combination of the replacement of activation functions and the temporal extension to the model. Here we refer to this kind of VGG-style neural networks as PlainNet, to make a distinction from those multi-branch networks such as ResNet and Inception.

We adopt the TDBN technique of Zheng et al. (2021) in our spiking models and formulate it as

\[ u_i^t = \tau_{\text{mem}} \cdot u_i^{t-1} + TDBN(I_i^t, \mu_{ci}, \sigma_{ci}^2, V_{th}), \]  

where \( \mu_{ci}, \sigma_{ci}^2 \) are channel-wise mean and variance calculated per-dimension over the sequence of mini-batches \( \{I_i^t\}_{t=1,...,T} \).

A direct transcription of the basic block from non-spiking ResNet is shown in Fig. 1, which is used in almost all of previous spiking ResNet works. A shortcut connection is inserted into PlainNet and turns it into its residual counterpart, PlainResNet. The shortcut connections mostly work as identity mapping, except that an additional \( 1 \times 1 \) convolution and BN would be utilized when downsampling is needed. The residual block can be written as

\[ o_i^t = \text{LIF}(\mathcal{F}(o_i^{t-1}) + o_i^{t-1}). \]  

where \( \mathcal{F}(\cdot) \) represents the group of nonlinear functions in a residual path, and \( i, l - 1 \) represent the current layer and the previous layer, respectively.

In view of the big obstacle caused by the limited network structures of SNNs, whether the degradation problem has been solved with BN and shortcut connections is a question worth asking. Therefore, we conduct experiments on CIFAR-10 (Krizhevsky and Hinton 2009) with varying depth, and it should be noted that our focus is on the degradation problem rather than pushing the state-of-the-art results, so we intentionally use deep but relatively narrow architectures as in Table 1. The results are shown in Fig. 2. The accuracy of PlainNet with BN begins to drop at the depth of 14 layers, and surprisingly the adoption of shortcut connections will just shift the peak to a depth of 20 layers. Despite a gentler slope after the peak, severe accuracy loss occurs when the depth reaches over 56 layers. The degradation problem still exists in spite of the introduction of BN and shortcut connections, making building a sufficiently deep SNN a nontrivial task.

| Layers | 1 + 2n | 2n | 2n |
|--------|--------|----|----|
| Output size | 32x32 | 16x16 | 8x8 |
| Channels | 16 | 32 | 64 |

Table 1: The structure for depth analysis on CIFAR-10.

The Crux of Degradation

It is the spiking activation function \( \text{LIF}(\cdot) \) between residual blocks that we consider as the crux of the degradation problem. Here we assume that the information loss might be caused by the inconsistency of two physical quantities in the shortcut connection and the residual path.

More specifically, there are spike trains \( o_i^{t-1,l} \) directly transmitted from the previous layer in the shortcut connections, whereas what the residual paths convey are normalized synaptic inputs \( \mathcal{F}(o_i^{t,l-1}) \), which are conceptually closer to the membrane potentials. The difference between the two quantities causes mismatch when they are added and sent to \( \text{LIF}(\cdot) \), since \( o_i^{t-1,l} \) lies in \{0, 1\} but \( \mathcal{F}(o_i^{t,l-1}) \) satisfies a continuous normal distribution. The mismatch indicates that if the residual block expects an instant change to \( o_i^{t,l-1} \) through learning, the residual path shall meet

\[ \begin{cases} \mathcal{F}(o_i^{t,l-1}) + \tau u_i^{t-1,l} < V_{th} - 1 & \text{if } o_i^{t-1,l} = 1 & \text{& } o_i^{t,l} \rightarrow 0 \\ \mathcal{F}(o_i^{t,l-1}) + \tau u_i^{t-1,l} > V_{th} & \text{if } o_i^{t-1,l} = 0 & \text{& } o_i^{t,l} \rightarrow 1. \end{cases} \]  

For \( \mathcal{F}(o_i^{t,l-1}) \) that does not meet the above conditions, it will not be able to affect the firing pattern of this layer. In other words, the information conveyed by the residual path will not be received by the next layer unless it is strong enough, which is in conflict with the original purpose of
residual learning. In ResNet, the residual path is set to learn the residual part to the identity mapping and the part is potentially small.

To verify the information loss resulted from the interblock \(\text{LIF}(\cdot)\), we adopt structural similarity index measure (SSIM) (\cite{Wang2004}) to quantify the changes between firing patterns. SSIM ranges from \(-1\) and \(+1\) and only equals \(+1\) if the two images are identical. The pixel-values of the firing pattern are defined as the firing rates in the rate-coding scheme.

If the firing pattern changes uniformly between layers, it will appear as a circle in the radar chart of Fig. 3. Although a strict circle does not seem to be applicable for the whole neural network due to encoding/voting or downsampling layers, our ResNet has shown a rounder curve than PlainResNet. Especially, there are 5 layers with an SSIM value of \(+1\) in PlainResNet-56, which implies that the firing patterns do not change when the information flows through the residual blocks and that these layers fail to help feature extraction. Consequently, a following layer needs to compensate for the inaction of preceding layers, so it is shown as an ensuing dramatic information change. The workload of the whole network is unbalanced, which undoubtedly reflects the irrationality of the existing PlainResNet structure when it comes to the deep network training.

An auxiliary experiment is designed to provide additional evidence. We remove the interblock \(\text{LIF}(\cdot)\) functions and maintain those in the residual paths as nonlinearity. The results are shown in Fig. 2. Within the depth of 56 layers, there is no loss of accuracy and the degradation problem has been effectively alleviated, so we mainly identify the crux of degradation problem as the interblock \(\text{LIF}(\cdot)\).

### Spiking Residual Blocks

As mentioned above, the design of spiking ResNet needs to consider the information loss caused by \(\text{LIF}(\cdot)\) between blocks, and make use of every residual block for giving full play to the deep structure. In this section, we introduce additional design criteria and propose several corresponding alternatives. We also analyze their potentials for tackling the degradation problem through depth analysis experiments on CIFAR-10. Finally, we manage to get a novel residual block, which is capable of training a 482-layer-deep network and no degradation problem is observed. Additional results on CIFAR-100 (\cite{Krizhevsky2009}) are available in Supplementary Material A.

#### Design Criteria

First, it should be pointed out that ResNet without interblock \(\text{LIF}(\cdot)\) is undesired for neuromorphic computing. One main source of energy efficiency for neuromorphic computing is...
the spike-based convolution, which means that the convolutional layer receives and processes the binary spiking input. Specialized optimization, for example the look up table, can be consequently applied to further boost its efficiency on neuromorphic devices (Liang et al. 2021). Once \textit{LIF}(\cdot) is removed, the CONV at the top of the next block will receive the continuous input rather than binary spikes, causing difficulty in benefiting from spike-based convolution and rich input/output sparsity. Therefore, we try to explore a novel residual block design which could resolve the degradation problem to expand the application scope as well as maintain the computing efficiency of SNNs:

- The structure should have enough capability to expand to a large scale while ensuring satisfying accuracy.
- In order to maintain the computing efficiency of SNNs, all convolutional layers (except the encoding layer and the downsampling layers) should receive and process spiking inputs.
- Not too much additional computing overhead should be caused by the novel structure for a fair comparison at the same network depth.

**Alternative Structures**

The proposed alternative residual blocks are presented in Fig. 4. Our solutions can be divided into two types according to whether to retain the interblock \textit{LIF}(\cdot). The retaining structure keeps the interblock \textit{LIF}(\cdot) whereas the removing structures move it into the residual path. According to the arrangement of the layers in the residual path, the removing type includes three different structures. Table 2 displays the depth analysis results on the structures investigated. The depth is the only variable and more experimental details can be found in Supplementary Material A.

**The Retaining Structure: ResNet\textit{RT}**. Since the mismatch between binary spiking signals and continuous membrane potentials impedes the information flow, we try to unify what flows in the residual path and shortcut connection by adding constant CONV$1 \times 1$ and BN in the shortcut connection and turning its signals into continuous membrane potentials. We do not consider converting the information of both paths into spiking signals, because this implies that the two paths will have the same importance, which is also inconsistent with the design purpose of ResNet. The model is named ResNet\textit{RT} for simplicity. As given in Table 2, in shallow structures, ResNet\textit{RT} outperforms PlainResNet adopted by previous work. When the network goes deeper, though ResNet\textit{RT} does not completely avoid degradation, it can alleviate the problem to a certain extent.

**The Removing Structures: ResNet\textit{RM}**. The other type of structures is based on the ResNet without interblock \textit{LIF}(\cdot) and reconstructs the residual path in the block (Fig. 4).

Unlike Eq.(6), the removing of the interblock \textit{LIF}(\cdot) can provide a clean forward path for the information flow, which can be written as

\[
\begin{align*}
  u_i^l &= u_i^{l-1} + F(\textit{LIF}(u_i^{l-1})), \quad (8) \\
  u_i^L &= u_i^1 + \sum_{l=2}^L F(\textit{LIF}(u_i^{l-1})). \quad (9)
\end{align*}
\]

In the backward pass, we have

\[
\begin{align*}
  \frac{\partial \text{Loss}}{\partial u_i^{l-1}} &= \frac{\partial \text{Loss}}{\partial u_i^l} \left( 1 + \frac{\partial F(\textit{LIF}(u_i^{l-1}))}{\partial u_i^{l-1}} \right). \quad (10) \\
  \frac{\partial \text{Loss}}{\partial u_i^1} &= \frac{\partial \text{Loss}}{\partial u_i^L} \left( 1 + \sum_{l=2}^L \frac{\partial F(\textit{LIF}(u_i^{l-1}))}{\partial u_i^l} \right). \quad (11)
\end{align*}
\]

Benefitted from the clean path throughout the whole network, inputs from previous layers can always be a component for later outputs and correspondingly a part of the gradient information will have a direct impact on the inputs. Therefore, this structure guarantees a smooth information flow and alleviates the difficulty in the training of deep networks.

It should be noted that in the removing structures, the encoding layer does not include a \textit{LIF}(\cdot) function to convert raw images to spiking inputs because the conversion will be carried out by the first residual path’s \textit{LIF}(\cdot) function.
In addition, an extra \( LIF(\cdot) \) is placed at the end of the entire convolutional part, to ensure that even if the information only flows through the shortcut connections, the subsequent classifier will still receive spiking signals.

From the depth analysis on CIFAR-10 (Table 3), despite the slight differences in the residual path, the three structures can all expand to a large scale without facing the degradation problem. This emphasizes the importance of a thorough identity mapping in the whole network, as in \cite{He et al. 2016a}. Rather than increasing accuracy by a slight margin on deep ANN structures, adopting such a design in SNNs results in significant improvements in both network scalability and task performance.

With regard to the selection between the three structures, we mainly take computing efficiency into our consideration. ResNet\(^{RM}\) requires mixed-precision addition for information aggregation and does not meet the previous spiking CONV criterion. ResNet\(^{RM}\) and ResNet\(^{RM}\) are close as options. The accuracy of ResNet\(^{RM}\) is a little behind and it causes a slightly heavier memory overhead. Besides, the separation of CONV and BN in ResNet\(^{RM}\) causes difficulty in utilizing the BN fusion technique that can simplify the structure for further acceleration optimization. Therefore, we finally take spiking ResNet\(^{RM}\) as our final choice, refer to it as SRM-ResNet and validate its effectiveness through further experiments.

Before carrying out experiments on ImageNet, we set \( n \) to 80 and obtain an extremely deep SRM-ResNet482 on CIFAR-10 with accuracy of 91.9%. Although the improvement in accuracy is not significant due to increasingly prominent overfitting problem, it surely evidences the scalability and the capability of our model to avoid the degradation.

| Depth | Alternatives | Plain ResNet |
|-------|--------------|--------------|
| 32    | 86.89        | 89.48        |
| 44    | 80.47        | 90.04        |
| 56    | 85.14        | 90.78        |
| 110   | N.A.         | 92.12        |

Table 2: Depth analysis of alternative structures.

### Experiments

#### ImageNet

**Experimental Setup.** We evaluate our models on the ImageNet 2012 dataset \cite{Deng et al. 2009}. The models are trained with the 128 million training images and tested with the 50k validation images. The training recipe at first simply follows that of \cite{He et al. 2016a} for our SRM-ResNet18 and SRM-ResNet34, i.e., a \( 224 \times 224 \) random crop with horizontal flip for data augmentation and a SGD optimizer with a weight decay of \( 1e-4 \) and a momentum of 0.9. However, the model would suffer from severe overfitting when it is extended to a depth of 104 layers. Despite a significant improvement on the training set, the testing accuracy only increases by a negligible margin. To fully unleash the potential of the deep spiking model SRM-ResNet104, an advanced training recipe is taken with stronger data augmentation and stronger regularization, and we adopt a fast \( T=1 \) pre-train phase before the time-consuming multi-timesteps formal training to save time (See Supplementary Material B for more training details).

**Results.** As the network deepens, satisfying accuracy increases have been achieved and great model scalability has been evidenced (Table 3). It shows the effectiveness of our \( LIF(\cdot) \) working in the residual paths as activation functions, despite spike-based computation required, and clearly demonstrates the potential of deepening SNN models.

When compared with other advanced works, our ResNet-34 with accuracy of 69.42% surpasses all previous directly trained SNNs and our ResNet-104 with accuracy of 74.21% is even comparable to the converted SNNs with much fewer time steps required for an inference. An accuracy gap to ANNs of about 2.6% is close to those of multi-timesteps converted SNNs and we also find that just enlarging the images for inference from \( 224 \times 224 \) to \( 288 \times 288 \) can improve the accuracy to a more competitive score at 76.02%.

#### DVS Datasets

**Experimental Setup.** CIFAR10-DVS is an event-stream dataset for object classification \cite{Li et al. 2017}. 10,000 frame-based images from CIFAR-10 are recorded by a dynamic vision sensor (DVS) and converted into event streams. We take the CIFAR10-DVS dataset to test the spatiotemporal processing capability of our models and adopt the data preprocessing in \cite{Fang et al. 2020}.

**Results.** Our ResNet-20 achieves a record on CIFAR10-DVS (Table 4). The model mainly follows the deep but narrow paradigm in Table 4 except that an additional down-sampling is placed at the first convolution stage due to the larger input size. Our deepest model actually has a small amount of parameters compared with other works, which is about one-sixth of those in \cite{Yao et al. 2021} and \cite{Fang et al. 2020}.

#### Energy Efficiency Estimation

Great sparsity is observed in our SRM-ResNet models. The firing rates of our spiking ResNet-34 and ResNet-104, defined as the firing probability of each neuron per time step, are 0.225 and 0.192, respectively. We notice a certain pattern appears in the alternate layers (Fig. 5). The second layer in each residual path always shows a higher firing rate (more apparent phenomenon for ResNet-104 can be seen in Supplementary Material C), which indicates that our residual block has a relatively active intermediate representation while the spiking activities after the confluence of two branches are kept calm and sparse.

One of the features intentionally maintained in our models is the spiking CONV, which could replace the multiply-and-accumulate (MAC) operations in ANNs with spike-driven synaptic accumulate (AC) operations in SNNs. To further demonstrate the energy efficiency, we estimate the energy cost based on the number of operations and the data for various operations in 45nm technology \cite{Horowitz 2014}.
| Method       | Work                        | Model     | Time step | Acc.(%) |
|--------------|-----------------------------|-----------|-----------|---------|
| ANN Conversion | (Han, Srinivasan, and Roy 2020) | VGG-16 | 4096      | 73.09   |
|              |                             | ResNet-34 | 2500      | 69.96   |
|              |                             | ResNet-50 | 350       | 71.61   |
|              |                             | EfficientNet-B7 | 3488 | 83.57   |
| Hybrid Training | (Hu, Tang, and Pan 2018) | ResNet-34 | 2500      | 72.75   |
|              |                             | ResNet-50 | 800       | 75.10   |
| Direct Training | (Rathi et al. 2020) | VGG-16 | 250       | 65.19   |
|              |                             | ResNet-34 | 600       | 61.48   |
|              |                             | ResNet-50 | 600       | 64.88   |
|              |                             | Wide-ResNet-34 | 600 | 67.04   |
|              |                             | ResNet-101 | 400       | 67.78   |
|              |                             | ResNet-104 | 400       | 68.76   |
| Direct Training | (Zheng et al. 2021) | ResNet-18 | 250       | 63.10   |
|              |                             | ResNet-34 | 600       | 64.42   |
|              |                             | ResNet-104 | 600       | 74.21   |
|              |                             | ResNet-104* | 600      | 76.02   |

Table 3: ImageNet results. †A spike is allowed to carry multi-bit information. ‡Since ResNet-104 is not a standard ResNet model, we train its ANN counterpart under the same recipe. *The input crops are enlarged to 288×288 in inference.

| Work          | Method          | Params | Acc. (%) |
|---------------|-----------------|--------|----------|
| (Ramesh et al. 2020) | DART        | N.A.   | 65.78   |
| (Kugel et al. 2020) | Rollout-ANN   | 0.5M   | 66.75   |
| (Zheng et al. 2021) | Spiking ResNet-19 | 12.6M  | 67.80   |
| (Yao et al. 2021) | TA-SNN        | 1.7M   | 72.00   |
| (Fang et al. 2021) | PLIF          | 1.5M   | 74.80   |
| Our work      | SRM-ResNet20   | 0.27M  | 75.56   |

Table 4: Results on CIFAR10-DVS.

| Model          | G-FLOPs | E(ANN) (1E-3J) | G-SyOPs | E(SNN) (1E-3J) |
|----------------|---------|----------------|---------|----------------|
| ResNet-34      | 3.53    | 16.22          | 4.77    | 4.29           |
| ResNet-104     | 11.79   | 54.24          | 11.32   | 10.19          |

Table 5: Convolutional energy consumption in the residual paths for a single ImageNet image.

Figure 5: Layer-wise numbers of spikes and firing rates of our SRM-ResNet34 (the last FC is not included). It is averaged on 100 randomly chosen images in the validation set.

We focus solely on convolutional layers in the residual paths, which constitute a major part of floating-point operations (FLOPs). The encoding layer and downsampling layers in the shortcuts do not meet the spiking CONV requirement, but their fixed cost only accounts for 4% of FLOPs in ResNet-34 and 1% in ResNet-104. With the sparsity of spikes and short simulation process, our SRM-ResNet can achieve the calculation of the residual part with about the same number of synaptic operations (SyOPs) rather than FLOPs (Table 5), which means that each neuron emits only one spike on average. As the network deepens, the ratio of energy consumption for a single image will approach a value of \( T \times \text{firing rate} \times \frac{E(\text{SNN})}{E(\text{ANN})} \), and notably the number of time steps \( T \) will not contribute as a factor when we deal with spatiotemporal datasets, so the energy efficiency will be more prominent (See Supplementary Material C for more details).

**Conclusion**

In this work, we identify the interblock information loss crux and propose a novel spiking residual block to tackle the degradation problem, which enables the direct training of a 482-layer model on CIFAR-10 and a 104-layer model on ImageNet. The great depth brings superior representation power. To our best knowledge, this is the first time such high performance is reported on ImageNet with directly trained SNNs. In addition, our resulting models attain vary sparse spiking activities and extremely low latency, indicating remarkable energy efficiency especially for spatiotemporal information processing. A deep and powerful SNN model is surely to work as the backbone for our further exploration in brain-inspired computing.
References

Abbott, L. F. 1999. Lapicque’s Introduction of the Integrate-and-Fire Model Neuron (1907). *Brain research bulletin*, 50(5-6): 303–304.

Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.-J.; Tab, B.; Beakes, M.; Breeze, B.; Kuang, J. B.; Manohar, R.; Risk, W. P.; Jackson, B.; and Modha, D. S. 2015. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neuromorphic Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10): 1537–1557.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chintya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; Liao, Y.; Lin, C.-K.; Lines, A.; Liu, R.; MathaiKutty, D.; McCoy, S.; Paul, A.; Tse, J.; Venkataramanan, G.; Weng, Y.-H.; Wild, A.; Yang, Y.; and Wang, H. 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1): 82–99.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Deng, L. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep Residual Learning in Spiking Neural Networks. arXiv:2102.04159.

Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; and Tian, Y. 2020. Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks. arXiv:2007.05785.

Han, B.; Srinivasan, G.; and Roy, K. 2020. RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity Mappings in Deep Residual Networks. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 630–645. Cham: Springer International Publishing. ISBN 978-3-319-46493-0.

Hodgkin, A. L.; and Huxley, A. F. 1952. A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. *The Journal of physiology*, 117(4): 500–544.

Horowitz, M. 2014. 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)*, 10–14.

Hu, Y.; Tang, H.; and Pan, G. 2018. Spiking Deep Residual Network. *arXiv preprint arXiv:1805.01352*.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Izhikevich, E. 2003. Simple Model of Spiking Neurons. *IEEE Transactions on Neural Networks*, 14(6): 1569–1572.

Kim, S.; Park, S.; Na, B.; and Yoon, S. 2020. Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07): 11270–11277.

Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images.

Kugele, A.; Pfeil, T.; Pfeiffer, M.; and Chicca, E. 2020. Efficient Processing of Spatio-Temporal Data Streams With Spiking Neural Networks. *Frontiers in Neuroscience*, 14: 439.

Lee, C.; Kosta, A. K.; Zhu, A. Z.; Chaney, K.; Daniilidis, K.; and Roy, K. 2020. Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 366–382. Cham: Springer International Publishing. ISBN 978-3-030-58526-6.

Li, H.; Liu, H.; Ji, X.; Li; and Shi, L. 2017. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11: 309.

Liang, L.; Qu, Z.; Chen, Z.; Tu, F.; Wu, Y.; Deng, L.; Li, G.; Li, P.; and Xie, Y. 2021. H2Learn: High-Efficiency Learning Accelerator for High-Accuracy Spiking Neural Networks. *arXiv preprint arXiv:2107.11746*.

Maass, W. 1997. Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural networks*, 10(9): 1659–1671.

Mayr, C.; Hoeppner, S.; and Furber, S. 2019. Spinnaker 2: A 10 million Core Processor System for Brain Simulation and Machine Learning. *arXiv preprint arXiv:1911.02385*.

Neftci, E. O.; Mostafa, H.; and Zenke, F. 2019. 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.

Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W.; Chen, F.; Deng, N.; Wu, S.; Wang, Y.; Wu, Y.; Yang, Z.; Ma, C.; Li, G.; Han, W.; Li, H.; Wu, H.; Zhao, R.; Xie, Y.; and Shi, L. 2019. Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture. *Nature*, 572(7767): 106–111.

Ramesh, B.; Yang, H.; Orchard, G.; Le Thi, N. A.; Zhang, S.; and Xiang, C. 2020. DART: Distribution Aware Retinal Transform for Event-Based Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11): 2767–2780.

Rathi, N.; Srinivasan, G.; Panda, P.; and Roy, K. 2020. Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation. In *International Conference on Learning Representations*.

Rueckauer, B.; Lungu, I.-A.; Hu, Y.; Pfeiffer, M.; and Liu, S.-C. 2017. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. *Frontiers in Neuroscience*, 11: 682.

Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; and Roy, K. 2019. Going Deeper in Spiking Neural Networks: VGG and Residual Architectures. *Frontiers in Neuroscience*, 13: 95.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Training Very Deep Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’15, 2377–2385. Cambridge, MA, USA: MIT Press.

Stöckl, C.; and Maass, W. 2021. Optimized Spiking Neurons Can Classify Images with High Accuracy through Temporal Coding with Two Spikes. *Nature Machine Intelligence*, 3(3): 230–238.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections in Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI’17, 4278–4284. AAAI Press.
Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going Deeper With Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Tarnowski, W.; Warchol, P.; Jastrzębski, S.; Tabor, J.; and Nowak, M. 2019. Dynamical Isometry is Achieved in Residual Networks in a Universal Way for any Activation Function. In Chaudhuri, K.; and Sugiyama, M., eds., Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89 of Proceedings of Machine Learning Research, 2221–2230. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, u.; and Polosukhin, I. 2017. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, 6000–6010. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Veit, A.; Wilber, M.; and Belongie, S. 2016. Residual Networks Behave like Ensembles of Relatively Shallow Networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16, 550–558. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE transactions on image processing, 13(4): 600–612.

Wu, J.; Yılmaz, E.; Zhang, M.; Li, H.; and Tan, K. C. 2020. Deep Spiking Neural Networks for Large Vocabulary Automatic Speech Recognition. Frontiers in Neuroscience, 14: 199.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; and Shi, L. 2018. Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. Frontiers in Neuroscience, 12: 331.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019. Direct Training for Spiking Neural Networks: Faster, Larger, Better. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01): 1311–1318.

Yang, Z.; Wu, Y.; Wang, G.; Yang, Y.; Li, G.; Deng, L.; Zhu, J.; and Shi, L. 2019. DashNet: A Hybrid Artificial and Spiking Neural Network for High-speed Object Tracking. arXiv preprint arXiv:1909.12942.

Yao, M.; Gao, H.; Zhao, G.; Wang, D.; Lin, Y.; Yang, Z.; and Li, G. 2021. Temporal-wise Attention Spiking Neural Networks for Event Streams Classification. arXiv preprint arXiv:2107.11711.

Zhang, M.; Wang, J.; Zhang, Z.; Belatreche, A.; Wu, J.; Chua, Y.; Qu, H.; and Li, H. 2020. Spike-timing-dependent back propagation in deep spiking neural networks. arXiv preprint arXiv:2003.11837.

Zheng, H.; Wu, Y.; Deng, L.; Hu, Y.; and Li, G. 2021. Going Deeper With Directly-Trained Larger Spiking Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12): 11062–11070.

Zhou, S.; Li, X.; Chen, Y.; Chandrasekaran, S. T.; and Sanyal, A. 2021. Temporal-Coded Deep Spiking Neural Network with Easy Training and Robust Performance. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12): 11143–11151.