A. Backpropagation Training with Surrogate Gradients

Following the previous surrogate gradients works [8,7,9,10,14,12,6,13], we train the weight parameters with surrogate gradients function in order to circumvent non-differentiable problem of LIF neurons. Specifically, the output spikes $o^t_i$ are generated if the membrane potential $u^t_i$ exceeds a firing threshold. Here, we use $arctan(\cdot)$ function following the previous work [3]:

$$\frac{\partial o^t_i}{\partial u^t_i} = \frac{1}{\pi} \arctan(\pi x) + \frac{1}{2}. \quad (1)$$

We follow the loss function $L$ used in [1] which accumulates cross-entropy loss across all timesteps. The gradients for each weight are computed in both spatial and time axis [14,9]. According to the chain rule, we can compute the gradients of the weight parameters $W_l$ as:

$$\frac{\partial L}{\partial W_l} = \sum_t \{ \frac{\partial L}{\partial O^t_l} \frac{\partial O^t_l}{\partial U^t_l} + \frac{\partial L}{\partial U^{t+1}_l} \frac{\partial U^{t+1}_l}{\partial U^t_l} \} \frac{\partial U^t_l}{\partial W_l}. \quad (2)$$

Here, $O^t_l$ and $U^t_l$ are the matrix form of output spikes and membrane potential at time-step $t$ for layer $l$, respectively.

B. Late Rewinding for SNNs

Frankle et al. [4] present Late Rewinding, which rewinds the network to the weights at epoch $i$ rather than initialization. This enables IMP to discover the winning ticket with less performance drop in high sparsity regime by providing a more stable starting point. As shown in Fig. 1, we found that Late Rewinding shows better performance than the original IMP in deep SNNs. Throughout our main paper, we apply Late Rewinding to IMP for experiments where we rewind the network to epoch 20.
C. Variation of KL Divergence at the Early Training Stage

We found that the normalized KL divergence metric (for each $T_{early}$) shows high variation in the first epoch, as shown in Fig. 2. After epoch 2, the standard deviation of the normalized KL divergence metric goes smaller, and the mean normalized KL divergence maintains a similar value across training. Therefore, we train SNN for 2 epochs in order to stabilize the KL divergence metric while minimizing training cost. Note, we can also select more higher number of epochs, however, they will provide the same $T_{early}$.

D. Time Cost for VGG and ResNet Architectures

To provide the reference on search time measured in the main paper, we show time cost for one training iteration (feedforward + backward) on VGG16 and ResNet19 in Table 1. We use SpikingJelly which provides optimized LIF neuron implementation on NVIDIA RTX 2080ti GPU and Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz processor. The execution time is proportional to the size of the backward computational graph [2] which is affected by timesteps. The results show the execution time exponentially increases if the computational graph
Table 1. Time cost for one iteration (feedforward + backward) with respect to the number of timesteps. We use batch size 128 for the experiments.

| Method          | Number of Timesteps |
|-----------------|---------------------|
|                 | T=1     | T=2     | T=3     | T=4     | T=5     |
| VGG16 [11]      | 26.45   | 29.15   | 32.44   | 37.26   | 44.49   |
| ResNet19 [5]    | 27.57   | 31.14   | 39.76   | 49.85   | 61.50   |

goes larger. Also, we observe that ResNet19 requires more execution time than VGG16, and the difference increase with higher timesteps.

Table 2. Ablation studies on a metric for $T_{early}$ selection. We use JSD and KLD for experiments. Note, the value at T=2 is fixed to 1 as a reference. We report mean and standard deviation from 5 runs.

| Method          | Jensen–Shannon Divergence | Kullback-Leibler Divergence |
|-----------------|---------------------------|----------------------------|
|                 | T=2 | T=3 | T=4 | T=2 | T=3 | T=4 |
| VGG16 [11]      | 1± 0.00 | 0.281± 0.028 | 0.167± 0.027 | 1± 0.00 | 0.274± 0.052 | 0.158± 0.063 |
| ResNet19 [5]    | 1± 0.00 | 0.375± 0.010 | 0.233± 0.021 | 1± 0.00 | 0.367± 0.031 | 0.243± 0.038 |

E. Experiments on Early Time (ET) tickets

In Table 3, we report the change in accuracy and search speed gain from applying ET to IMP and EB, on SVHN, Fashion-MNIST, CIFAR10, and CIFAR100 datasets. The results support our hypothesis that important weight connectivity of the SNN can be discovered from shorter timesteps.

F. Applying Various Distance Metrics

To discover $T_{early}$, we use Kullback-Leibler Divergence (KLD) between the class probabilities across different timesteps (refer Algorithm 1 in the main paper). In this section, we measure the normalized Jensen–Shannon Divergence (JSD), a symmetric version of KLD, in Table 2. We can observe that both KLD and JSD show a similar distance for VGG16 and ResNet19 architecture. This implies that our $T_{early}$ algorithm is robust to the distance metric.

G. Experiments on deep/shallow networks

To further validate the existence of LTH in SNNs, we ran the experiments with ResNet34 on TinyImageNet. In Fig.3 Left, we found that Winning Ticket exists even in deep networks on larger dataset with various pruning methods including IMP, EB, IMP+ET, and EB+ET. We also report a random pruning as a baseline, which implies Winning Ticket is difficult to be discovered with a naïve approach. We also evaluate AlexNet on CIFAR10 in Fig.3 Right. The results corroborate our statement in which LTH can be applied to SNNs.
Fig. 3. Accuracy comparison of pruning methods on two settings. **Left:** ResNet34 on TinyImageNet. **Right:** AlexNet on CIFAR10. We show mean/standard deviation from 5 random runs.

References

1. Deng, S., Li, Y., Zhang, S., Gu, S.: Temporal efficient training of spiking neural network via gradient re-weighting. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=_XNtisL32jv
2. Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., Tian, Y., other contributors: Spikingjelly. https://github.com/fangwei123456/spikingjelly (2020)
3. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2661–2671 (2021)
4. Frankle, J., Dziugaite, G.K., Roy, D.M., Carbin, M.: Stabilizing the lottery ticket hypothesis. arXiv preprint arXiv:1903.01611 (2019)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
6. Kim, Y., Li, Y., Park, H., Venkatesha, Y., Panda, P.: Neural architecture search for spiking neural networks. arXiv preprint arXiv:2201.10355 (2022)
7. Lee, C., Sarwar, S.S., Panda, P., Srinivasan, G., Roy, K.: Enabling spike-based backpropagation for training deep neural network architectures. Frontiers in Neuroscience 14 (2020)
8. Lee, J.H., Delbruck, T., Pfeiffer, M.: Training deep spiking neural networks using backpropagation. Frontiers in neuroscience 10. 508 (2016)
9. Neftci, E.O., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks. IEEE Signal Processing Magazine 36, 61–63 (2019)
10. Shrestha, S.B., Orchard, G.: Slayer: Spike layer error reassignment in time. arXiv preprint arXiv:1810.08646 (2018)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015)
12. Wu, H., Zhang, Y., Weng, W., Zhang, Y., Xiong, Z., Zha, Z.J., Sun, X., Wu, F.: Training spiking neural networks with accumulated spiking flow. ijo 1(1) (2021)
13. Wu, J., Xu, C., Zhou, D., Li, H., Tan, K.C.: Progressive tandem learning for pattern recognition with deep spiking neural networks. arXiv preprint arXiv:2007.01204 (2020)
Table 3. Effect of the proposed Early-Time ticket. We compare the accuracy and search time of Iterative Magnitude Pruning (IMP), Early-Bird (EB) ticket, Early-Time (ET) ticket on four sparsity levels. We show search speed gain and accuracy change from applying ET. Note, ResNet19 requires 1.06 ~ 1.38× forward + backward time compared to VGG16 because of a large computational graph from multiple timesteps.

| Setting Method | CIFAR10 | SVHN | FMNIST | CIFAR100 | Res19 | VGG16 | CIFAR10 | SVHN | FMNIST | CIFAR100 | Res19 | VGG16 |
|----------------|---------|------|--------|----------|-------|-------|---------|------|--------|----------|-------|-------|
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |