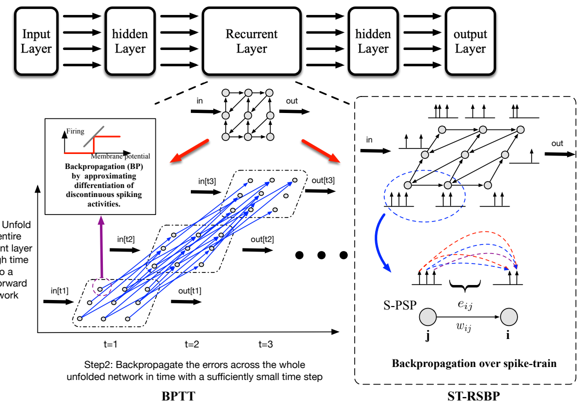
We deeply appreciate and will address all insightful review comments in final paper with major ones responded below:

1. Main contributions: ST-RSBP can transparently train all types of SNNs including RSNNs without unrolling in time. The employed S-PSP model improves training efficiency at the spike-train level and also addresses discontinuity of spiking activity for accurate gradient computation. The spike-train level processing for RSNNs is the starting point for ST-RSBP. After that, we have applied the standard BP principle while dealing with specific issues of derivative computation at the spike-train level. Unlike methods such as Feedback Alignment, Direct Feedback Alignment, and e-prop, ST-RSBP is not biologically plausible - a limitation. Biologically plausible methods tend to produce somewhat lower performance; ST-RSBP trades off biological plausibility for performance.

2. Comparison with BPTT: We revise Figure 2 on the right to better illustrate the difference between the proposed ST-RSBP and other BPTT based rules. BPTT first unfolds a RSNN in time to effectively remove recurrent connections and then backpropagates the error across the whole unfolded network and along the discretized time points, during which non-differentiability of spiking activity must be dealt with. ST-RSBP operates on the spike-train level, preforms training while avoiding unfolding the RSNN and backpropagating through the long unfolded path.



3. Comparison with other works: Table 1 compares the proposed ST-RSBP with other works on N-MNIST, the well known neuromorphic version of MNIST. Since none of these works has been tested on RSNNs, we only compare the results on feed-forward SNNs. As shown in Table 1, ST-RSBP outperforms the BPTT based rules [23, 38] and is slightly better than [19, 33]. Moreover, ST-RSBP is readily applicable to RSNNs. In Table 2, we evaluate ST-RSBT using the Sequential MNIST dataset under the RSNN setting based on the same preprocessing of [4]. The LIF network in [4] is a fully-connected RSNN without the special adaptive neurons proposed in [4] and is trained using BPTT. We test ST-RSBP on fully-connected RSNNs with a size equal to or smaller than that of the LIF network. Table 2 shows that the proposed ST-RSBP outperforms the BPTT adopted in [4]. Note that the main contribution of [4] is on a new type of SNNs, namely Long Short-Term Memory Spiking Neural Networks (LSNNs) with the special adapting neurons, demonstrating very good performance. Our comparison here only intends to show that from a training perspective, ST-RSBP outperforms BPTT when training similar standard RSNNs. We expect that by modifying our ST-RSBP rule we can also train LSNNs to enhance training quality.

Table 1: Performance on N-MNIST

Model	Hidden layers	Accuracy
Spiking MLP[23]	800	98.74%
STBP[38]	800	98.78%
HM2BP[19]	800	98.88%
SLAYER[33]	500-500	98.89%
ST-RSBP	800	**98.91%**

Table 2: Performance on Sequential MNIST

Model	Hidden layers	Accuracy
LIF[4]	R220	63.30%
ST-RSBP	R128	**76.52%**
ST-RSBP	R220	**77.39%**

4. Event-based Processing, Hardware-Friendliness, Implementation Settings, and Fashion MNIST

ST-RSBP performs supervised training and only updates the weights at the end of the spike train of each example when the loss is available. However, the computation of S-PSPs, the main overhead fo ST-RSBP, can be accumulated spike-by-spike in an event-driven online manner in the forward pass of BP, removing the need of storing the spiking history of the network. This feature makes ST-RSBP amenable to neuromorphic hardware implementation. Recently, we have successfully demonstrated online S-PSP computation on FPGA for a different training algorithm.

The parameters like the desired output firing counts, thresholds, learning rates are empirically tuned. The chosen values for each network reported in Section 4 are summarized in the "Results" directory of the source code repository.

Non-spiking ANNs that produce better results than ST-RSBP on Fashion-MNIST are different types of CNNs. We do not evaluate ST-RSBP on spiking CNNs. In Table 4, we only compare ST-RSBP with the best performing methods on non-CNN feedforward networks including BP for ANNs. Training spiking CNNs using BP is very time consuming. In the future, we will demonstrate the application of ST-RSBP to spiking CNNs.

For demonstration purpose, we adopt the recurrent model for Fashion MNIST to show the ability of ST-RSBP to train RSNNs on different datasets. Here we also use ST-RSBP to train a 400-400 feed-forward SNN with accuracy of 90.08% on Fashion MNIST, also surpassing all other methods in Table 4 of the paper.