Incorporating temporal spike stability into the Forward-Forward algorithm

Alexandru-Mihai Radu

Supervisor(s): Stephanie Tan, Yaqi Guo

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

June 21, 2026

Name of the student: Alexandru-Mihai Radu
Final project course: CSE3000 Research Project
Thesis committee: Inald Lagendijk, Stephanie Tan, Yaqi Guo

Incorporating temporal spike stability into the Forward-Forward algorithm

Alexandru-Mihai Radu
Delft University of Technology

Abstract

Recent interest in biologically plausible alternatives to backpropagation has renewed attention on Spiking Neural Networks and the Forward-Forward algorithm, where learning is driven by local layer-wise goodness functions rather than global error gradients. In most Forward-Forward learning implementations of Spiking Neural Networks, goodness is defined as spike-count activity, leaving temporal properties of neural activity unused. This work investigates whether temporal spike stability, measured using the inter-spike interval coefficient of variation (ISI-CV), can improve Forward-Forward learning in fully connected leaky integrate-and-fire spiking neural networks. Using MNIST as a benchmark, we evaluate several ISI-CV-based extensions, including direct temporal penalties, contrastive gap losses, plasticity based approaches, and candidate scoring. Directly optimizing for temporal regularity conflicts with the Forward-Forward goodness margin and destabilizes training. The use of ISI-CV as a plasticity control signal, that reduces updates to temporally stable neurons, can be used to fine-tune the model. ISI-CV-based candidate scoring performs above chance, indicating that spike timing contains class-related information, but remains weaker than standard goodness-based classification.

1 Introduction

Recent advances in power-efficient neural architectures have strengthened the position of Spiking Neural Networks (SNNs) as a promising alternative to conventional artificial neural networks (ANNs) [1]. Unlike standard ANNs, which typically process information through continuous values, SNNs communicate through discrete spike events distributed over time. This temporal and discontinuous nature introduces additional challenges for learning. In particular, applying conventional backpropagation-based training to SNNs is mathematically less direct than in ANNs, since spike events are non-differentiable and neural activity evolves over time [1].

At the same time, interest in biologically plausible learning has motivated the development of alternatives to backpropagation that rely on local learning rules. One notable example is the Forward-Forward (FF) algorithm [2], which was originally designed to train ANNs. It replaces the backward pass with layer-wise learning based on goodness functions evaluated on positive and negative data. This approach has shown promising results in terms of biological plausibility and computational efficiency, while avoiding some of the limitations associated with backpropagation [2].

FF's local learning rule motivates its adaptation to SNNs [3]. SNNs provide a biologically inspired computational substrate, while FF provides a learning rule that is more local and avoids a conventional backward pass.

The central issue in this adaptation then becomes the design of the goodness function itself, as goodness functions designed for ANNs do not use any of the SNNs temporal properties [4]. This problem becomes more apparent when we take into consideration that the goodness function used in current implementations of FF learning for SNNs is based on the sum of squared (SoS) neuron activations [3]. SoS functions not only do not include temporal properties of SNNs, but they also fail to capture structural properties of neural activity already present in ANNs and SNNs [5, 4]. This issue has been largely mitigated for ANNs through large-scale studies of shape-aware and selectivity-based goodness functions that have shown that structural properties can improve performance [6, 5]. In the case of SNNs this remains an open problem for both structural and temporal properties.

More recently, the Inter-Spike Interval Coefficient of Variation (ISI-CV) has been used effectively in SNNs to extract neuron importance information to counteract the effect of catastrophic forgetting in online learning [7]. The study showed that temporally stable neurons can encode information relevant to classification and that ISI-CV can be used to effectively extract that information. Moreover, in biological neuroscience, ISI-CV has been used to extract cortical patterns and classify neural activity states [8]. This raises a natural question: can temporal stability, extracted using ISI-CV, be leveraged to improve FF learning and classification accuracy when using SNNs?

We investigate ISI-CV integrations in three categories. First, building on the ANN goodness-function study [6], we modify the FF objective, either by directly augmenting the goodness function or by adding an auxiliary contrastive loss. Second, using [7] as a starting point, we use ISI-CV as a plasticity-control[1] signal by updating neuron weights according to temporal stability. Third, inspired by shape-based approaches [5], we evaluate ISI-CV during classification through class-specific temporal fingerprints used for candidate scoring.

Our contributions are as follows: (1) we present an analysis into the existing temporal stability properties of Forward-Forward trained Spiking Neural Networks; (2) we present six experiments that incorporate temporal spike stability via ISI-CV into Forward-Forward learning on Spiking Neural Networks.

The remainder of this paper is organized as follows: Section 2 introduces the theoretical background into the Forward-Forward algorithm, its implementation in Spiking Neural Networks and ISI-CV. Section 3 presents our network and hyperparameter setup together with the mathematical background behind the six experiments presented. Section 4 shows our results and their interpretation. Section 5 summarizes our findings and presents our directions for future work in this field. Section 6 contains information regarding the usage of LLMs, use of prior work and reproducibility.

2 Background

2.1 Forward-Forward in Spiking Neural Networks

The Forward-Forward (FF) algorithm, introduced by Hinton [2], is an alternative to backpropagation in which each layer is trained using local objectives rather than gradients propagated backward through the full network. During training, instead of one forward pass followed by one backward pass, FF uses a *positive pass* followed by a *negative pass*. Positive pass input data is constructed by pairing the input sample with its corresponding correct label, while negative pass input is constructed by pairing the input sample with an incorrect label. FF uses a function called *goodness function* to measure how correct a label is for an input sample. The FF's objective during training is for each layer to assign higher *goodness* to positive data and lower *goodness* to negative data. During classification FF uses this property to run the input against all possible labels to choose

[1] Plasticity control refers to modulating how strongly a neuron's weights are updated during training.

the label with the highest *goodness*. As such, the design of the *goodness function* directly influences the performance and behavior of FF [6].

Spiking neural networks (SNNs) differ from conventional artificial neural networks because they communicate through discrete spike events over time rather than continuous values. This gives SNNs a closer connection to biological neural systems and allows information to be represented through both firing rate and spike timing. A common neuron model used in SNNs is the leaky integrate-and-fire (LIF) neuron. In this model, the membrane potential integrates incoming input over time and emits a spike when it crosses a firing threshold. After a spike is produced, the membrane potential is reset or reduced.

In many implementations of FF using SNNs, the goodness function is defined using the magnitude of neural activity, such as the squared activation or spike-count activity [3]. As a result, most FF implementations on SNNs ignore potentially useful temporal information. This limitation is relevant because temporal structure is one of the defining characteristics of spiking neural networks.

2.2 Inter-Spike Interval Coefficient of Variation

In biological neuroscience, *Inter-Spike Interval Coefficient of Variation* (ISI-CV) and related temporal activity features are commonly used to characterize neural firing patterns, distinguish between different activity states, and analyze changes in cortical dynamics [8]. Since spiking neural networks are designed to be more biologically plausible [9], such a metric may also provide useful information for analyzing artificial spike trains and understanding how information is represented over time.

The advantage of ISI-CV is the ability to compare temporal stability of neurons with different firing rates. It achieves this through normalization of inter-spike variability $\sigma(ISI)$ by the mean inter-spike interval $\mu(ISI)$

$$CV_{ISI} = \frac{\sigma(ISI)}{\mu(ISI)}. \tag{1}$$

A lower ISI-CV indicates more regular and temporally stable spike timing, while a higher ISI-CV indicates more irregularity. More information about the mathematical notations used throughout this paper can be found in Appendix E.

3 Methodology

We use a fully connected spiking neural network trained with the Forward-Forward algorithm as the base for our experiments Figure 1. The network receives the dataset samples at the input layer and processes them through two hidden layers of leaky integrate-and-fire (LIF) neurons. The input to the network consists of flattened MNIST images with 784 dimensions. The hidden layers contain 500 neurons.

LIF Neuron Model We use the LIF neuron model because it provides a simple and widely used abstraction of neuron dynamics in SNNs [3, 10, 11]. For neuron i at timestep t, the membrane potential $u_i[t]$ retains part of its previous state $u_i[t-1]$ through a decay factor while integrating all received input at timestep t. The discrete-time membrane potential update $u_i[t]$ used in this work can be written as

$$u_i[t] = \beta u_i[t-1] + \sum_{j=1}^{N} w_{j,i} s_j[t], \tag{2}$$

where β is the membrane decay factor, N is the size of the previous layer, $w_{j,i}$ is the synaptic weight given to the connection between the neuron j and neuron i, and $s_j[t]$ is the spike emitted by neuron j at timestep t.

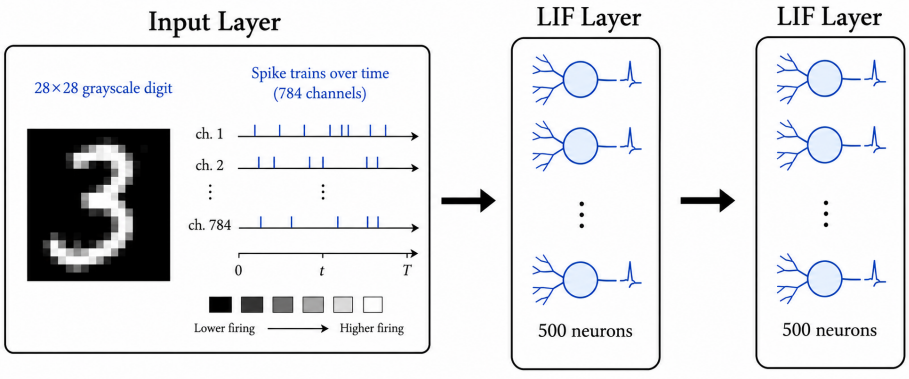


Figure 1: Network architecture containing the input layer and the two LIF layers. *Lower firing* and *higher firing* refer to the relation between image pixel values and the firing rate at the output of the input layer (Eq. 7). (AI generation was used)

The spike output is generated by applying a threshold function to the membrane potential $u_i[t]$:

$$s_i[t] = \begin{cases} 1, & \text{if } u_i[t] \geq V_{\text{th}}, \\ \text{otherwise.} & 0, \end{cases} \tag{3}$$

The spike function (Eq. 3) as defined above is not differentiable and gradients cannot be computed from it [12]. To avoid this problem during gradient computation, the derivative of the spike function is replaced by a fast-sigmoid surrogate [13, 12].

Baseline FF goodness The baseline Forward-Forward goodness function we use is based on the sum of squared spike-count activity [3]. The goodness function of a layer h can be written as

$$G(h) = \frac{1}{N} \sum_{i=1}^{N} C(h_i)^2 \quad \text{with} \quad C(h_i) = \sum_{t=1}^{T} h_i[t], \tag{4}$$

where $C(h_i)$ counts the number of spikes from a given spike train[2], h_i is the spike train of a neuron i, N is the number of neurons in the layer.

The goodness function is used to compute the *goodness gap* (Eq. 5) between positive and negative samples during training and the candidate score of a sample-label pair during classification.

$$G_\Delta = G(h^{pos}) - G(h^{neg}) = G^{pos} - G^{neg} \tag{5}$$

Baseline FF loss For our baseline FF implementation we keep the original objective of increasing the gap G_Δ. To achieve this, we use the swish SiLU loss function directly in the loss computation (Eq. 6) [3].

$$\mathcal{L} = \text{SiLU}(-\alpha(G_\Delta)). \tag{6}$$

[2]The output spike record of a neuron is also referred to as a spike train.

Dataset The dataset used is MNIST, which consists of grayscale digit images belonging to ten classes, from 0 to 9. Each image has a spatial resolution of 28×28 pixels. Before being passed to the network, each image is flattened into a one-dimensional vector, of size 784. We include label information directly in the input representation as presented in [3].

MNIST is a non-spiking dataset that requires an encoding scheme in order to transform it into usable spike trains for the SNN. We use Poisson rate encoding [14] as it has a higher degree of biological plausibility compared to the static encoding seen in [3]. In Poisson encoding, each input value controls the probability of emitting a spike at each timestep (Figure 1). For input neuron i, the output spike at timestep t is generated by

$$s_i[t] \sim \text{Bernoulli}(r_i) \quad \text{with} \quad r_i = \text{clip}(\alpha x_i, 0, 1), \tag{7}$$

where r_i is the firing probability associated with pixel value x_i. Thus, brighter pixels have a higher probability of producing spikes, while darker pixels produce fewer or no spikes.

3.1 ISI-CV

The *inter-spike interval coefficient of variation* (ISI-CV) is computed using the spike trains extracted during forward passes. For each neuron, we compute its ISI-CV 1 using the spike train generated by one input sample over the T simulation window. Spike trains are not concatenated over multiple samples to avoid destroying sample-specific information.

ISI-CV can be computed reliably when a neuron produces enough spikes to form multiple inter-spike intervals. In our work, a neuron is considered valid for ISI-CV computation only when it produces at least 3 spikes during the T simulation window of a given sample. This gives at least two inter-spike intervals under the standard definition. Neurons with fewer than 3 spikes are assigned a value of $CV_{ISI} = 3$. More details regarding the parameters of ISI-CV are provided in Appendix A.

Algorithm 1 Surrogate ISI-CV computation for N neurons. Layer abstractions and data clipping are excluded.

Require: Spike tensor $S \in [0, 1]^{T \times N}$
Ensure: Differentiable ISI-CV score $y \in \mathbb{R}^N$
Notation: $\odot$ denotes the Hadamard product.

$a \leftarrow \mathbf{0}_N$
$I \leftarrow [\,]$
for $t = 1$ **to** T **do**
  Append a to I
  $a \leftarrow (a + 1)(1 - S_t)$ ▷ Set age to 0 if we have a spike, add 1 otherwise.
end for
$D \leftarrow \sum_{t=1}^{T} S_t$ ▷ Compute number of spikes per neuron.
$\mu \leftarrow \dfrac{\sum_{t=1}^{T} S_t \odot I_t}{D}$ ▷ Compute average ISI length.
$\sigma^2 \leftarrow \dfrac{\sum_{t=1}^{T} S_t \odot (I_t - \mu)^2}{D}$ ▷ Compute ISI variance.
$\text{CV} \leftarrow \dfrac{\sqrt{\sigma^2}}{\mu}$ ▷ Compute ISI-CV.
return CV

Surrogate ISI-CV Compared to similar studies where ISI-CV is used only for statistics [7], we conduct experiments where ISI-CV is part of the training objective. As ISI-CV is based on spikes, it is not differentiable,

creating a similar issue to the one encountered by the LIF threshold function (Eq. 3). To solve this problem we replace the normal ISI-CV computation by a differentiable surrogate (Algorithm 1).

When the ISI-CV metric is numerically combined with another metric, such as squared activation, an adaptive scale factor ρ is used for scaling (Eq. 8, 10). Without it, the temporal stability term may either become too small or too large and overwhelm computations.

3.2 Baseline Model and Hyper-parameters

All experiments use as starting point the same baseline model and hyper-parameters. This ensures that differences observed between experiments are caused by the modifications themselves. Moreover, the baseline temporal behavior that forms the basis of our experiments is presented in Appendix B.

The hyper-parameters and the baseline model training used are summarized in Table 1 from SectionA. Specific experiment parameters, such as penalty strength λ, temporal margin m, fingerprint size K, or soft-freezing minimum contribution $p_{\min}$, are reported separately in the corresponding experimental results sections.

3.3 Experimental Setups

We present the six experiments which integrate temporal stability into the FF algorithm.

The six experiments are organized around three progressively weaker hypotheses about the role of ISI-CV in FF-trained SNNs. The strongest hypothesis is that temporal stability can be directly incorporated into the FF training objective in order to improve performance. This is tested through the positive-sample goodness augmentation, symmetric goodness augmentation, and contrastive gap loss. The second hypothesis is that ISI-CV can improve performance by selecting the plasticity-control of neurons during training. This is tested through hard freezing and soft freezing. The weakest hypothesis is that temporal stability contains class-specific information even if it is not suitable as a direct training objective or plasticity-control signal. This is tested through candidate scoring with class-specific temporal fingerprints.

All experiments use the same dataset, input encoding, network architecture, LIF parameters and optimizer. Because ISI-CV is unstable with random spike trains, we start the experiments on an already trained network and test whether the network's performance improves under the new rules. We therefore compare the mean final accuracy of each method against the mean accuracy of the starting checkpoint, which we refer to as the baseline (97.68%). More information can be found in Appendix A.

Positive-Sample Goodness Augmentation Our first experiment incorporates ISI-CV into the existing Forward-Forward learning objective through the goodness function. The experiment is based on the hypothesis that correct input-label pairs should not only produce high spike-count goodness as required by the FF objective, but may also produce more temporally stable spike trains. If this assumption holds, then encouraging lower ISI-CV for positive samples could improve the quality of the learned representation and therefore the accuracy of the model. The direct augmentation into the goodness function through a factor λ (Eq. 8) borrows from the goodness function research into ANNs [6].

Let $G(h^{pos})$ denote the standard FF goodness calculation for a positive sample and $G(h^{neg})$ for a negative sample. In this experiment total positive goodness G^{pos} is modified by subtracting an ISI-CV penalty, while the total negative goodness G^{neg} remains unchanged. The total positive goodness function can be written as

$$\widetilde{G}^{pos} = G(h^{pos}) - \lambda\rho\overline{CV}_{ISI}^{pos} \tag{8}$$

where λ is a weighting coefficient controlling the strength of the temporal penalty, ρ is the adaptive scaling factor used to match the magnitude of the ISI-CV term to the standard goodness term and $\overline{CV}_{ISI}^{pos}$ is the mean ISI-CV value computed from the positive sample spike trains.

By replacing G^{pos} in Eq. 5 with the value from Eq. 8 we can write the new goodness gap as

$$\widetilde{G}_{\Delta} = \widetilde{G}^{pos} - G^{neg} = \underbrace{(G(h^{pos}) - G(h^{neg}))}_{G_{\Delta}} - \lambda\rho\overline{CV_{ISI}^{pos}} \tag{9}$$

This formulation encourages the model to preserve the standard Forward-Forward separation between positive and negative samples G_{Δ}, while applying pressure on the optimizer to reduce temporal irregularity in the positive sample response. The optimizer will try to decrease the value of $\overline{CV}_{ISI}^{pos}$ in order to increase the value of $\widetilde{G}_{\Delta}$.

Symmetric Goodness Augmentation Our second experiment extends the first experiment by applying the temporal stability penalty to both positive and negative samples. By doing this, the optimizer has to increase the gap between $\overline{CV}_{ISI}^{pos}$ and $\overline{CV}_{ISI}^{neg}$. If, during an update, improving ISI-CV for $\widetilde{G}^{pos}$ is easy while for $\widetilde{G}^{neg}$ is hard, it could decide to improve only one of them, mainly $\widetilde{G}^{pos}$ in order to increase the gap between them. This experiment directly aligns the ISI-CV gap objective with Forward-Forward's goal of increasing the goodness gap between positive and negative labels.

$$\begin{aligned} \widetilde{G}^{pos} &= G(h^{pos}) - \lambda\rho\overline{CV_{ISI}^{pos}} \\ \widetilde{G}^{neg} &= G(h^{neg}) - \lambda\rho\overline{CV_{ISI}^{neg}} \end{aligned} \tag{10}$$

where $\overline{CV_{ISI}^{pos}}$ and $\overline{CV_{ISI}^{neg}}$ denote the mean ISI-CV values computed from positive and negative sample activity, respectively. As in the previous variant (Eq. 8), λ controls the strength of the temporal term and ρ is an adaptive scaling factor used to align the magnitude of the ISI-CV term with the standard goodness term. The λ and ρ are shared for $\widetilde{G}^{pos}$ and $\widetilde{G}^{neg}$. The resulting goodness gap is

$$\widetilde{G}_{\Delta} = \widetilde{G}^{pos} - \widetilde{G}^{neg} = \underbrace{(G(h^{pos}) - G(h^{neg}))}_{G_{\Delta}} + \lambda\rho\underbrace{(\overline{CV_{ISI}^{neg}} - \overline{CV_{ISI}^{pos}})}_{CV_{ISI\Delta}} \tag{11}$$

Compared to the positive-sample variant, this formulation is more explicitly contrastive and more similar to the goodness functions described in similar studies, where both positive and negative passes are included [6]. It assumes that the difference between positive and negative temporal stability is meaningful and can be used for improving the model's accuracy.

Contrastive Gap Loss The third experiment integrates ISI-CV as a separate objective, keeping the baseline FF loss (Eq. 6) unchanged. To accomplish this, we add a new *hinge loss* term, similarly to [15, 16, 6], that enforces a margin between the ISI-CV values of positive and negative samples.

In studies [17–19], auxiliary training objectives have been used successfully to improve the performance of the main task. Thus, objective separation may mitigate objective alignment issues between FF and ISI-CV. The standard Forward-Forward loss is still responsible for increasing the goodness gap between positive and negative samples, while the contrastive ISI-CV term independently encourages a temporal stability gap.

The desired temporal relation is similar to the previous one in Eq. 11 and can be written as $\overline{CV_{ISI}^{neg}} - \overline{CV_{ISI}^{pos}} \geq m$, meaning that positive samples should produce more temporally stable spike trains than negative samples by an amount m. Thus we define the *hinge loss* term as follows

$$\mathcal{L}_{CV} = \mathrm{SiLU}(m + \underbrace{\overline{CV_{ISI}^{pos}} - \overline{CV_{ISI}^{neg}}}_{-CV_{ISI\Delta}}). \tag{12}$$

The *hinge loss* approaches zero only when $CV_{ISI\Delta}$ is larger than the margin m. Our final training objective can be written as

$$\mathcal{L} = \mathcal{L}_{FF} + \lambda \mathcal{L}_{CV}, \tag{13}$$

where $\mathcal{L}_{FF}$ is the standard Forward-Forward loss and λ controls the strength of the contrastive ISI-CV penalty, similar to [20, 17].

Hard Freezing In this experiment we want to test the behavior of ISI-CV as a plasticity-control signal rather than as a direct objective. Our motivation stems from the successful use of ISI-CV and similar metrics in the prevention of catastrophic forgetting [7, 21]. These methods are tested on networks with conceptually different objectives and learning mechanisms from FF [7], and it is worth exploring their interaction with FF. Freezing temporally stable neurons follows the intuition that regular spike timing may indicate stable learned representations, but it has not been shown yet to be different than freezing temporally unstable neurons.

In our implementation we use ISI-CV to decide which neurons should be *frozen* from further updates during training. This *freezing* of neuron updates is done by zeroing weight gradients associated with specific neurons.

We define the weight update as follows

$$w_{j,i} \leftarrow w_{j,i} - \eta \, z_i \frac{\partial \mathcal{L}}{\partial w_{j,i}}, \quad \text{with} \quad z_i = \begin{cases} 0, & i \in F, \\ 1, & \text{otherwise.} \end{cases} \tag{14}$$

where $w_{j,i}$ is the weight for the connection between neuron j and neuron i, $\mathcal{L}$ is the loss function, η is the learning step, z_i is the frozen neuron mask and F the set of frozen neurons.

Soft Freezing We extend the hard freezing experiment by replacing binary neuron freezing with a continuous weighting mechanism. In the hard freezing variant, selected neurons are either fully frozen or fully trainable. This can be too strict, because a neuron selected for freezing completely loses plasticity, while all non-frozen neurons remain fully plastic. The soft freezing variant avoids this binary decision by assigning each selected neuron a plasticity factor based on its ISI-CV rank.

Compared to hard freezing, soft freezing provides a smoother intervention. The formula used assigns linearly increasing values in the interval $[p_{min}, 1]$. The neurons with the lowest ISI-CV receive the lowest plasticity factors while temporally unstable neurons receive higher plasticity factors. We therefore define the plasticity factor f_i and our weight update rule as follows

$$\begin{gathered} f_i = p_{\min} + (1 - p_{\min}) \frac{\operatorname{rank}(v_i)}{n-1}, \quad \text{with} \quad \operatorname{rank}(v_i) \in \{0, 1, \ldots, n-1\} \\ w_{j,i} \leftarrow w_{j,i} - \eta \, f_i \frac{\partial \mathcal{L}}{\partial w_{j,i}}, \end{gathered} \tag{15}$$

Inactive neurons are assigned $f_i = 1$. Using this formula temporally unstable neurons and neurons that spike the least keep their weights plastic as they receive a plasticity factor closer to 1.

Candidate Scoring In this experiment we evaluate whether temporal stability can be used during classification. After training, each class may have a distinctive temporal stability pattern. If this pattern is specific enough for each class, then candidate labels may be classified using the temporal behavior of class-specific neuron groups.

For each MNIST digit class, we compute the average temporal stability of each neuron over a number of positive samples. These values represent the fingerprints of the digit classes from which we select a subset F_q of neurons of size K starting with the lowest ISI-CV neurons.

The fingerprint for a class q can be computed as follows:

$$F_q = \text{BottomK}\left(\sum_{a=1}^{A} CV_{ISI,a}^{pos}\right), \tag{16}$$

where A is the number of samples and K is the number of selected neurons per class.

Each input image is evaluated under all candidate labels, with the *candidate score* based directly on temporal regularity. For each image-label pair (x, q), we compute the average ISI-CV over the selected neuron set F_q

$$cs(x, q) = \frac{1}{K} \sum_{n \in F_q} CV_{ISI,n}^{(x,q)}. \tag{17}$$

The predicted label $\hat{y}$ is the candidate with the lowest $cs(x, q)$ candidate score.

$$\hat{y}(x) = \arg\min_q cs(x, q). \tag{18}$$

3.4 Evaluation Metrics

The primary evaluation metric is classification accuracy on the test set. Accuracy measures the fraction of test samples for which the predicted label $\hat{y}_i$ matches the true label y_i. Accuracy is used as the main measurement in order to answer the central research question of classification improvement. Accuracy data collected for the methods is directly compared to the accuracy of the baseline model at the start of training (97.68%).

Training loss is the second evaluation metric and is recorded during the experiments. For the baseline model, this corresponds to the standard Forward-Forward loss $\mathcal{L}_{FF}$. For method variants that add ISI-CV terms, the recorded loss includes the corresponding modified objective. This allows the training dynamics between ISI-CV-based models to be compared.

Lastly, we also record the valid spiking fraction (%), in order to evaluate how each experiment affects spiking in the network. Changes in the valid fraction would indicate changes in spiking frequency in the network and also a subsequent drop in computable ISI-CV neurons.

4 Experimental Results & Discussion

4.1 Positive-Sample Goodness Augmentation

The sweep shows that the positive sample method becomes unstable as the temporal penalty is increased, as shown in Figure 2. With $\lambda = 0.01$, the model initially drops from 97.68% to 95.78% after the first fine-tuning epoch, and only partially recovers to 97.04% by the end. With $\lambda = 0.015$, the effect becomes more significant as accuracy falls to around 93.41% after the first epoch and ends at 92.76%. For $\lambda = 0.02$ and $\lambda = 0.04$, the positive ISI-CV penalty collapses the learned structure, ending at 52.84% and 21.24%, respectively. The degradation shows the destructive interference of the ISI-CV term to the FF goodness objective.

The loss behavior during training is also consistent with the interpretation that the ISI-CV term degrades the FF objective instead of helping it Figure 2. The loss increases, with the effect becoming stronger for larger λ,

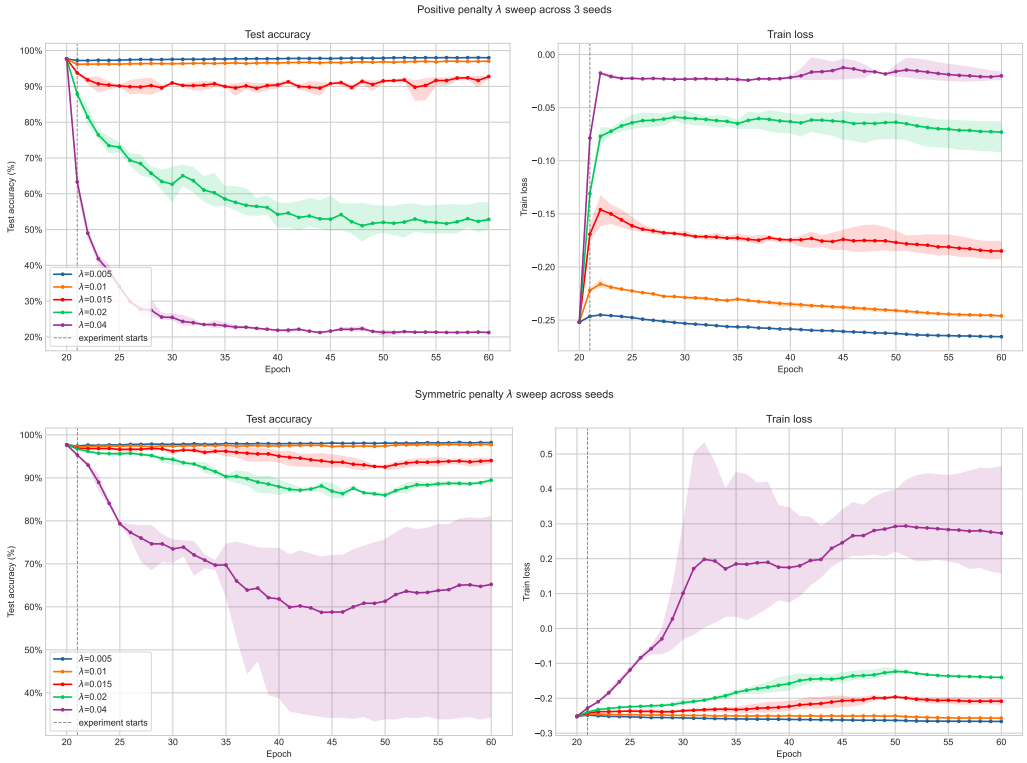


Figure 2: Accuracy and train loss for the positive-sample goodness augmentation and symmetric goodness augmentation, referred to here as *positive penalty* and *symmetric penalty*, across the λ sweep. Per-seed graphs can be found in Appendix D.1, D.2.

in which the ISI-CV term dominates the update direction. For small λ, this ISI-CV pressure is weak enough that the FF classifier can recover and even slightly improve over the 40 epochs.

The valid neuron fraction statistics further support the objective misalignment (Table 4). As λ increases, the fraction of neurons satisfying the minimum spike-count requirement decrease substantially, from a valid fraction of 0.01004 at $\lambda = 0.005$ to only 0.00133 at $\lambda = 0.04$. Thus, stronger positive ISI-CV penalties appear to suppress and destabilize the spiking activity needed for both ISI-CV estimation and FF classification.

4.2 Symmetric Goodness Augmentation

The symmetric ISI-CV goodness augmentation is less destructive than the positive penalty, but still shows a clear dependence on the strength of the temporal term (Figure 2). Like in positive sample experiment, the smallest temporal weight, $\lambda = 0.005$, gave the best result with a final accuracy of 98.2%. The decrease in the effect of ISI-CV on the weight update is likely to be caused by its application to both the positive and negative terms. This finding aligns well with the initial hypothesis that the temporal stability gradients might be reduced in comparison with the positive sample experiment.

The loss values support the same interpretation. For the two smallest penalties, the final loss remains close to or better than the initial loss, and classification accuracy remains high. In contrast, for $\lambda \geq 0.015$, the loss becomes progressively worse during training, reaching a positive final value for $\lambda = 0.04$.

Compared to the previous experiment, the valid neuron fraction, shown in Table 5, increases on with λ. Unlike the positive-only penalty, where large λ reduced the number of valid spiking neurons, the symmetric variant increases the valid fraction strongly as λ grows, from 0.01338 at $\lambda = 0.005$ to 0.15718 at $\lambda = 0.04$. This suggests that symmetric goodness augmentation fails by increasing spiking activity in a way that no longer preserves class-discriminative structure.

4.3 Contrastive Gap Loss

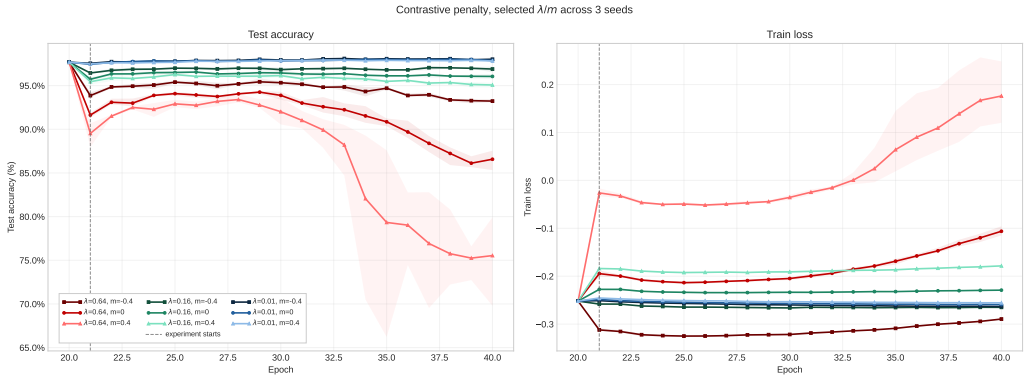


Figure 3: Accuracy and train loss for the contrastive gap loss, referred to here as the contrastive penalty, across the values of $\lambda \in \{0.1, 0.16, 0.64\}$ and $m \in \{-0.4, 0, 0.4\}$. Per-seed graphs and tables can be found in Appendix D.3.

The results in Figure 3 show that separating the objectives can prevent immediate collapse, but that enforcing a strong positive temporal margin between positive and negative samples is still poorly aligned with the standard FF objective.

With negative margins, the model remains mostly stable around the accuracy of the baseline model (97.68%). For $m = -0.4$, the final accuracies are 97.68%, 96.9%, and 93.23% for $\lambda = 0.04$, $\lambda = 0.16$, and $\lambda = 0.64$, respectively. Moreover, the relaxed margin seems to make the model stabilize more quickly after the initial drop in accuracy. A negative margin relaxes the desired condition $\overline{CV}_{ISI}^{neg} - \overline{CV}_{ISI}^{pos} \geq m$, so the auxiliary objective does not strongly force positive samples to become more temporally stable than negative samples.

For positive margins, the model consistently decreases in accuracy. The degradation is consistent with the previous experiments (Sections 4.2, 4.1). This can also be seen as a drop in the valid fraction statistics from Table 6.

Overall, directly optimizing for lower ISI-CV is not aligned with the FF goodness objective. For sufficiently large λ, the ISI-CV term destabilizes training rather than improving accuracy. The valid-fraction statistics suggest that this failure occurs through changes in network spiking activity, either by suppressing valid spiking, as in the positive-only and contrastive case, or by increasing activity in the symmetric case.

4.4 Hard Freezing

The hard-freezing experiment gives strong evidence that ISI-CV can be useful as a plasticity-control signal. Two tails are compared: freezing the lowest-ISI-CV neurons, which preserves the most temporally regular units, and freezing the highest-ISI-CV neurons, which preserves the most temporally irregular units.

Low ISI-CV neurons tend to contain more relevant information and protecting them is less harmful (Figure 4). Freezing the most temporally stable neurons preserves the trained classifier and allows the remaining neurons to continue adapting. The best final result is obtained with a 15% low-CV freeze, reaching 98.33% final accuracy and 98.34% best accuracy, above the baseline of 97.68% (Figure 5). Even when the freeze fraction is increased to 60%, the model still ends at 98.01%. This indicates that low ISI-CV neurons may form a stable part of the learned representation that can be protected[3] during continued FF training without harming performance.

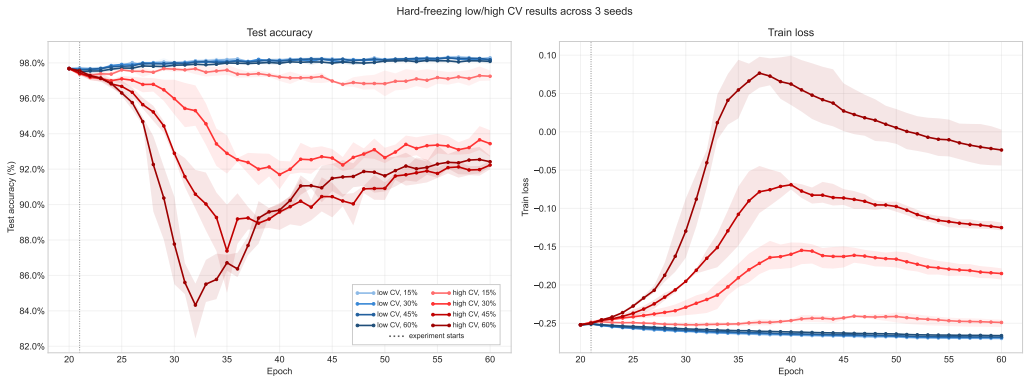


Figure 4: Test accuracy over the freezing of $\{low, high\}$ CV_{ISI} and over the frozen fraction size $\{15\%, 30\%, 45\%60\%\}$.

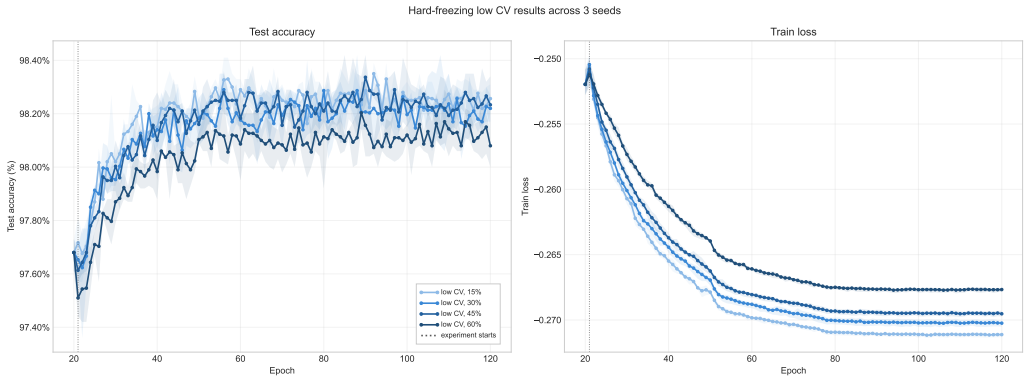


Figure 5: Test accuracy over the freezing of low CV_{ISI} and over the frozen fraction size $\{15\%, 30\%, 45\%60\%\}$. Per-seed graphs can be found in Appendix D.4.

The high ISI-CV freezing condition causes a drop in accuracy. Freezing only 15% of the high-CV neurons gives a final accuracy of 97.57%, slightly below the initial model. However, increasing the frozen high-CV fraction causes a large degradation: 30% high-CV freezing ends at 93.42%, 45% ends at 92.46%, and 60% ends at 92.12%. Freezing a larger portion of the unstable high-CV tail forces the model to modify temporally stable neurons, which seems to drop accuracy and reduce the spiking in the network (Table 7).

[3]Neuron protection refers to reducing or preventing updates to the weights associated with specific neurons in order to preserve their learned behavior.

4.5 Soft Freezing

Figure 6: Sweep of accuracy on test set and train loss over the $p_{min} \in \{0.05, 0.35, 0.65, 0.85\}$. More complete graphs can be found in Appendix D.5.

The soft-freezing experiment is the most stable of the ISI-CV-based training modifications. All soft-freezing variants improve over the baseline value of 97.68% (Figure 6). After soft freezing, final accuracies lie in a narrow range from 98.23% to 98.38% (Table 8). The differences between settings are small, which suggests that the method is not highly sensitive to the exact value of p_{min} in this range.

The training curves of the loss (Figure 6) support the interpretation that soft freezing preserves the FF optimization objective. The train loss decreases smoothly for all p_{min} values, from approximately -0.252 at the starting checkpoint to around -0.270 at the end.

Overall, the main conclusion from the two plasticity control experiments is that ISI-CV can be useful as a plasticity-control signal, especially when it is used to identify and protect stable neurons.

4.6 Candidate Scoring

The candidate-label ISI-CV scoring results in Figure 7 show that temporal stability is sufficiently class-specific to support classification even without modifying the trained FF-SNN. The most important observation is the strong dependence on the fingerprint size K. With only one selected neuron per class, the temporal readout already achieves 73.3% accuracy. With an increase in K we can achieve significant performance, rising by 20% from $K = 1$ to $K = 40$. The largest gains occur early, from 73.3% at $K = 1$ to 88.4% at $K = 5$, after which the improvement continues more gradually.

ISI-CV presents strong class biases as the per-class results show that the improvement with larger fingerprints is not driven by a single digit (Figure 7). At $K = 1$, digit 0 is already classified at 98.2%, but several classes remain much weaker, especially digit 1 at 53.5%, digit 6 at 66.4%, and digit 5 at 68%. By $K = 40$, all classes exceed 91% accuracy, with digit 5 the lowest at 91.7% and digit 0 the highest at 98.1%. Digit 1 shows the most striking improvement, increasing from 53.5% to 93.1%.

While increasing K seems to bring accuracy closer to baseline this result should be interpreted carefully, as the ISI-CV score also contains implicit spiking information. The experiment assigns an invalid ISI-CV value of 3 when the selected neurons do not meet the minimum spike requirement. Since classification chooses the candidate with the lowest mean ISI-CV, these assigned values of 3 correlate with candidate labels with less spiking in their selected fingerprint neurons. Consequently, as K increases, our score becomes more diluted by this spiking.

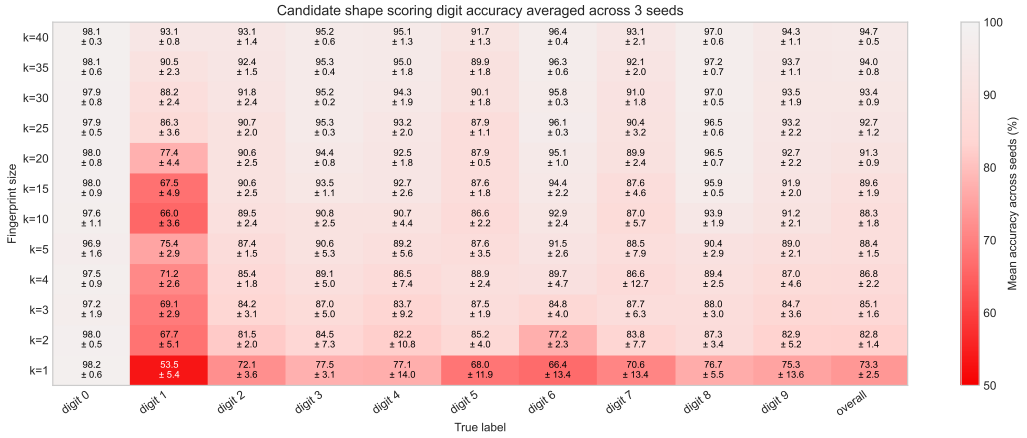


Figure 7: Candidate scoring accuracy per digit over the size of the fingerprint K. Per-seed tables can be found in Appendix D.6.

Overall, even when considering low values of K, the results of candidate scoring experiment clearly show that temporal stability does encode class-specific information and can be used for classification purposes.

5 Conclusions and Future Work

Our work investigated whether temporal spike stability, measured using the inter-spike interval coefficient of variation (ISI-CV), can improve Forward-Forward learning in spiking neural networks. The central hypothesis was that the temporal structure of spike trains contains useful information that is not captured by existing Forward-Forward SNN methods.

Directly optimizing for lower ISI-CV, either by modifying the goodness function or by adding a contrastive gap loss, does not reliably improve FF learning of SNNs. In most cases, these penalties destabilize training and reduce classification accuracy. Plasticity-control experiments show that weight updates based on ISI-CV can improve performance in FF learning of SNNs. Moreover, results suggest that temporally stable neurons are likely to encode learned representations when compared to unstable neurons. Finally, the candidate scoring experiment, shows that FF trained SNNs encode class-specific information in their temporal stability.

In conclusion, temporal spike stability of SNNs trained using Forward-Forward can provide information about how neurons encode learned representations and can help improve the performance of the model, but is not suited as an auxiliary objective during training.

Future work should further investigate the temporal structure of SNNs trained with the Forward-Forward algorithm by considering alternative temporal metrics beyond ISI-CV. In addition, future studies should evaluate approaches on datasets with richer temporal dynamics, including spike-based datasets, where temporal information is inherently central to the classification task. Finally, further experiments are needed to determine whether the observed usefulness of ISI-CV as a plasticity-control signal and candidate-scoring mechanism generalizes beyond the MNIST benchmark and the fully connected LIF architecture used in this work.

6 Responsible Research

Use of LLM-Assisted Code Generation Large language models were used during the development of code and the analysis of scripts. Their assistance was used as a programming and drafting aid, for example to help resolve code errors, improve helper methods, and support the review of analysis scripts. Generative AI was also used as a writing aid for this report, including grammar checking, style improvement, assistance with LaTeX formatting, and generating part of the architecture image. Generated suggestions and code were thoroughly reviewed, edited where necessary, and approved by the author. Generative AI was not used to produce the research question, the experimental design, the interpretation of the results, or any core aspect of the model used in our research.

Code Provenance and Use of Prior Work The baseline implementation is based on the Forward-Forward spiking neural-network approach described by [3] and hosted at `https://github.com/mngh-cs/SNN_FF/tree/main`. Their work provides an important starting point for combining the Forward-Forward algorithm with spiking neural networks. Our research extends the previous work with temporal spike metrics, ISI-CV analysis, several ISI-CV-based training experiments, and plasticity-control variants.

Dataset Use The experiments are conducted on MNIST, a standard publicly available dataset of handwritten digit images. The dataset used was provided by *pytorch* python library `https://docs.pytorch.org/vision/main/generated/torchvision.datasets.MNIST.html`. MNIST is widely used for benchmarking machine-learning models and does not contain sensitive information in the context of this work. The dataset is used only for training, classification experiments and diagnostic analysis of internal spike behavior.

At the same time, MNIST is a limited benchmark. It is relatively small, visually simple, and not representative of many real-world classification settings. Therefore, the results should not be interpreted as evidence that the proposed methods generalize to other domains.

Reproducibility All files required to reproduce the experiments are provided as *Interactive Python Notebooks* that can be run using any modern Python distribution with the following libraries installed PyTorch, snnTorch, torchvision, NumPy, pandas, and Matplotlib. Moreover, all used experimental settings are documented, including the dataset, model architecture, number of timesteps, LIF parameters, optimizer, learning rate, batch size, random seed, and training schedule. The code files can be accessed at `https://github.com/stan-group/alexandru_radu`.

Limitations The number of training and fine-tuning epochs was limited by the available time and computational resources. This caused some experiments to have shorter number of epoch especially when large hyper-parameter searches were done. As a result, some methods may not have reached their best possible performance, and longer runs may change the observed behavior of the ISI-CV-based variants.

The hyper-parameters were not exhaustively optimized. Parameters such as the learning rate schedule, firing threshold and membrane decay were taken from previous studies and were not optimized in any way.

References

[1] Rongguang Ye, Chenhao Ye, Chao Huang, Ming Tang, and Yunhao Liu. Beyond-backpropagation training: Methods, applications, and perspectives. *TechRxiv*, 2026(0103), 2026. doi: 10.36227/techrxiv.176740426.63642005/v1. URL `https://www.techrxiv.org/doi/abs/10.36227/techrxiv.176740426.63642005/v1`.

[2] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations, 2022. URL `https://arxiv.org/abs/2212.13345`.

[3] Mohammadnavid Ghader, Saeed Reza Kheradpisheh, Bahar Farahani, and Mahmood Fazlali. Backpropagation-free spiking neural networks with the forward-forward algorithm, 2025. URL `https://arxiv.org/abs/2502.20411`.

[4] Chenxiang Ma, Xinyi Chen, Yanchen Li, Qu Yang, Yujie Wu, Guoqi Li, Gang Pan, Huajin Tang, Kay Chen Tan, and Jibin Wu. Spiking neural networks for temporal processing: Status quo and future prospects, 2025. URL https://arxiv.org/abs/2502.09449.

[5] Talha Ruzgar Akkus, Suayp Talha Kocabay, Kamer Ali Yuksel, and Hassan Sawaf. Selectivity and shape in the design of forward-forward goodness functions, 2026. URL https://arxiv.org/abs/2604.13081.

[6] Arya Shah and Vaibhav Tripathi. In search of goodness: Large scale benchmarking of goodness functions for the forward-forward algorithm, 2025. URL https://arxiv.org/abs/2511.18567.

[7] Samrendra Roy, Kazuma Kobayashi, Souvik Chakraborty, Sajedul Talukder, and Syed Bahauddin Alam. Gradient-free continual learning in spiking neural networks via inter-spike interval regularization, 2026. URL https://arxiv.org/abs/2604.16496.

[8] Zilin Wang, Ao Li, Guihua Xiao, Xingzheng Gu, Zhilei Wang, Shuting Guo, Rujin Zhang, Chaowei Zhuang, and Jiangbei Cao. Unbiased segmentation and multifeature classification of cortical neuronal activity reveals complex dynamics under anesthesia. *Neuroscience Bulletin*, 42(3):491–504, 2026. ISSN 1995-8218. doi: 10.1007/s12264-025-01527-9. URL https://doi.org/10.1007/s12264-025-01527-9.

[9] Ali Safa. Continual learning in bio-plausible spiking neural networks with hebbian and spike timing dependent plasticity: A survey and perspective. *arXiv e-prints*, pages arXiv–2407, 2024.

[10] Sijia Lu and Feng Xu. Linear leaky-integrate-and-fire neuron model based spiking neural networks and its mapping relationship to deep neural networks, 2022. URL https://arxiv.org/abs/2207.04889.

[11] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection, 2019. URL https://arxiv.org/abs/1903.06530.

[12] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018. ISSN 1530-888X. doi: 10.1162/neco_a_01086. URL http://dx.doi.org/10.1162/neco_a_01086.

[13] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. 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, 2019. doi: 10.1109/MSP.2019.2931595.

[14] Daniel Auge, Julian Hille, Etienne Mueller, and Alois Knoll. A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters*, 53(6):4693–4710, 2021. ISSN 1573-773X. doi: 10.1007/s11063-021-10562-2. URL https://doi.org/10.1007/s11063-021-10562-2.

[15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815–823. IEEE, 2015. doi: 10.1109/cvpr.2015.7298682. URL http://dx.doi.org/10.1109/CVPR.2015.7298682.

[16] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18, 2005.

[17] Ang Li, Huiyi Hu, Piotr Mirowski, and Mehrdad Farajtabar. Cross-view policy learning for street navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8100–8109, 2019.

[18] Georgios Papoudakis, Kyriakos C. Chatzidimitriou, and Pericles A. Mitkas. Deep reinforcement learning for doom using unsupervised auxiliary tasks, 2018. URL https://arxiv.org/abs/1807.01960.

[19] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks, 2016. URL https://arxiv.org/abs/1611.05397.

[20] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. URL https://arxiv.org/abs/1705.07115.

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL `http://dx.doi.org/10.1073/pnas.1611835114`.

[22] Paulo Ricardo Protachevicz, CA Bonin, KC Iarosz, Iberê Luiz Caldas, and AM Batista. Large coefficient of variation of inter-spike intervals induced by noise current in the resonate-and-fire model neuron. *Cognitive Neurodynamics*, 16(6):1461–1470, 2022.

A Hyper-parameters

Table 1: Experimental setup and main hyper-parameters.

Category	**Hyperparameter**	**Value**
Architecture	Model type	Forward-Forward SNN
	Layer connectivity	Fully connected
	Input size	784
	Hidden size	500
	Number of hidden layers	2
	Dataset	MNIST with 60K/10K train/test split
Neuron and encoding	Input encoding	Poisson rate coding
	Rate-coding scale	1.0
	Neuron model	Leaky integrate-and-fire, with no biasing
	Membrane decay	0.99
	Firing threshold	1.5
	Simulation timesteps	20
Optimization	Optimizer	Adam
	Surrogate gradient	Fast-sigmoid surrogate
	Surrogate slope	25
	Seeds used	$\{35, 42, 49\}$
Learning-rate schedule	Epochs < 50	8×10^{-4}
	Epochs 50–74	3×10^{-4}
	Epochs 75–99	1×10^{-5}
	Epochs ≥ 100	1×10^{-6}
Forward-Forward objective	Loss function	Swish loss
	Swish parameter	$\alpha = 6.0$
Baseline model	Goodness function	Squared spike-count activity
	Number of epochs	20
	Seed accuracies	$\{97.73\%, 97.69\%, 97.61\%\}$
	Mean accuracy	97.68%
ISI-CV	Minimum spike count	3
	Value assigned to non-valid neurons	3

Most hyperparameters in Table 1 follow the experimental configuration introduced by [3] and its accompanying implementation at `https://github.com/mngh-cs/SNN_FF`. In particular, the network architecture, LIF neuron model,

goodness function, loss formulation, and learning rate schedule are based on this reference implementation. The number of simulation timesteps was inspired by [7] and was further validated empirically in our experiments (Figure 8). We observed that increasing the number of timesteps within a reasonable range did not improve or degrade the ISI-CV metric. Therefore, we fixed the number of timesteps to 20 for all reported experiments.

Biasing We do not use learned neuron biases in the LIF layers. A learned bias adds a constant input current to the membrane potential at every timestep, independently of the incoming spike activity. As a result, neurons may become more likely to fire even when little or no input is present during the simulation window. This independent spiking can confound temporal metrics such as ISI-CV, since the measured spike stability may partly reflect the induced excitability of the neuron rather than the temporal structure generated by the input sample.

Surrogate gradient and loss functions The LIF neuron emits a binary spike when the membrane potential crosses the firing threshold (Eq. 3). This thresholding operation is non-differentiable, as its derivative is zero almost everywhere and undefined at the threshold.

To address this issue, we use a surrogate-gradient approach. In the forward pass, the neuron still emits hard binary spikes, preserving the discrete spiking dynamics of the SNN. In the backward pass, however, the derivative of the threshold function is replaced by a smooth surrogate derivative. This is a standard approach for training SNNs, since it allows gradient-based optimization while retaining event-based spike communication in the forward computation [13, 12].

In our implementation, we use the fast-sigmoid surrogate gradient, similar to the implementation of [3]. The fast sigmoid provides a smooth approximation centered around the firing threshold [13], so that neurons whose membrane potentials are close to threshold receive the largest gradient signal. This is desirable because such neurons are the most sensitive to small changes in synaptic weights. At the same time, the gradient decays for membrane potentials far from threshold, reducing the influence of neurons whose spike state would be unlikely to change.

The baseline Forward-Forward objective is optimized using a Swish/SiLU loss applied to the negative goodness gap. This choice gives a smooth objective for increasing the separation between positive and negative samples. When G_Δ is large and positive, the loss approaches zero, meaning that the positive sample already has sufficiently higher goodness than the negative sample. When G_Δ is small or negative, the loss remains active and produces gradients that encourage the model to increase the goodness of positive samples relative to negative ones. The parameter α controls the sharpness of this transition.

ISI-CV The minimum number of spikes required for a valid ISI-CV computation follows from the interpretation of Eq. 1, and in particular from the $\sigma(\mathrm{ISI})$ term. Since $\sigma(\mathrm{ISI})$ measures the dispersion of inter-spike intervals, it requires at least two ISI samples to be meaningfully estimated. Under the standard biological neuroscience interpretation [22], a spike train containing n spikes produces $n-1$ ISIs. Therefore, obtaining two ISIs requires at least three spikes within the time window T.

This criterion differs from some computational implementations, including the implementation used in prior work [7], where the interval between the start of the time window and the first spike is also included in the ISI set. This convention effectively treats the initial membrane state as if it were preceded by a spike immediately before the beginning of the observation window. Under this interpretation, two emitted spikes can yield two intervals, making a minimum threshold of two spikes computationally valid but sensitive to boundary effects and isolated spike timing fluctuations. For this reason, in our analysis we compute with the extra interval, but use the stricter validity rule of least 3 spikes.

For neurons that do not meet this validity criterion, the assigned ISI-CV value is set to the minimum required spike count, namely 3. This choice provides a clear separation between invalid ISI-CV values and temporally unstable neurons. This contrasts with prior work that used a value of 2 for the same purpose [7].

B Baseline Temporal Behaviour

We first characterize the baseline temporal behavior of SNNs trained with the Forward-Forward algorithm, using the implementation described by [3] and the hyper-parameters listed in Table 1. This baseline analysis evaluates the temporal dynamics of the unmodified FF-SNN and provides a reference point for the ISI-CV-based variants introduced in our paper.

Neurons with fewer than three spikes are excluded from the mean; the valid columns report the percentage of neuron-sample entries with enough spikes (≥ 3) to compute ISI-CV.

Table 2: Mean ISI-CV on all valid neurons for 40 class-0 samples under the correct and wrong labels. Seed used is 42.

Label	Condition	Mean ISI-CV	Valid (%)	Mean spike count	Goodness
0	Correct	0.125853	1.900	0.15690	1.13010
1	Wrong	0.122774	1.480	0.12670	0.89810
2	Wrong	0.107030	1.465	0.12745	0.86345
3	Wrong	0.116913	1.395	0.12080	0.85690
4	Wrong	0.120561	1.455	0.12645	0.90085
5	Wrong	0.111232	1.560	0.13140	0.87000
6	Wrong	0.119008	1.470	0.12360	0.88740
7	Wrong	0.117112	1.545	0.12835	0.89295
8	Wrong	0.106046	1.500	0.12675	0.84555
9	Wrong	0.115816	1.440	0.12700	0.89880

Table 3: Mean ISI-CV on a subset of valid neurons for 40 class-0 samples under the correct and wrong labels. The subset size used is 50 and the seed used is 42.

Label	Condition	Mean ISI-CV	Valid (%)	Mean spike count	Goodness
0	Correct	0.068279	0.148	0.5260	1.13095
1	Wrong	0.080056	0.130	0.8290	0.89490
2	Wrong	0.076390	0.126	0.7880	0.85960
3	Wrong	0.086035	0.128	0.8165	0.85050
4	Wrong	0.084659	0.122	0.8720	0.89570
5	Wrong	0.077068	0.150	0.7470	0.87285
6	Wrong	0.083660	0.124	0.8235	0.87925
7	Wrong	0.086691	0.128	0.8680	0.88070
8	Wrong	0.089861	0.134	0.8000	0.84130
9	Wrong	0.074069	0.126	0.8500	0.89335

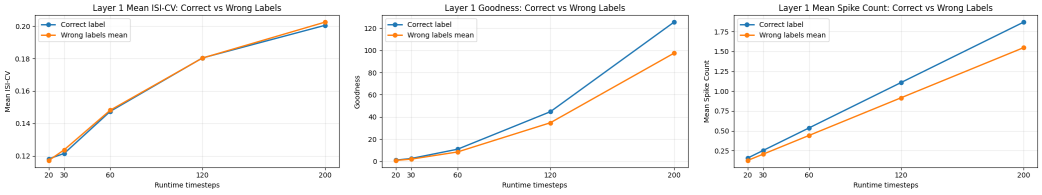


Figure 8: Gap between correct and wrong labels when considering mean ISI-CV, goodness and spike count.

The results from Table 2 show that averaging ISI-CV over all valid neurons does not produce a clear separation between correct and wrong candidate labels. The correct label has the highest goodness and a higher mean spike count than all wrong labels, confirming that the baseline Forward-Forward model separates labels mainly through spike-count activity. However, several wrong labels obtain lower mean ISI-CV values than the correct label. This indicates that a naive mean over all valid ISI-CV values does not provide a reliable temporal gap between correct and incorrect labels.

The timestep sweep in Figure 8 supports this interpretation. As the number of simulation timesteps increases, both goodness and spike count show a clearer separation between correct and wrong labels. In contrast, the ISI-CV gap remains small and does not show a consistent advantage for the correct label. This suggests that the standard FF SNN objective naturally strengthens spike-count separation, but does not automatically produce a global temporal stability separation.

Selecting a temporally stable subset exposes a gap between correct and wrong labels that is otherwise hidden. When ISI-CV is computed only on a selected subset of low-CV valid neurons, as shown in Table 3, the correct label obtains the lowest mean ISI-CV. The correct label reaches a mean ISI-CV of 0.068279, while all wrong candidate labels have higher values. When all valid neurons are averaged together, informative low-CV neurons may be diluted by neurons whose temporal variability is weakly related or unrelated to the candidate label.

Overall, standard FF SNN does not globally organize correct labels into lower ISI-CV responses, but temporally stable neuron subsets do contain label-dependent structure. This means that ISI-CV may not be sufficient as a replacement for the spike-count goodness function, but it can still provide useful information when extracted through neuron selection or candidate-specific temporal fingerprints.

C Tables

Table 4: Valid neuron fraction for the positive-sample augmentation λ sweep. Values are mean $\pm$ standard deviation across seeds 35, 42, and 49.

λ	Final accuracy (%)	Final valid neuron fraction (%)
0.005	98.00 ± 0.10	0.01004 ± 0.00029
0.010	97.04 ± 0.25	0.00924 ± 0.00016
0.015	92.76 ± 0.27	0.00832 ± 0.00048
0.020	52.84 ± 4.20	0.00314 ± 0.00085
0.040	21.24 ± 0.26	0.00133 ± 0.00069

Table 5: Valid neuron fraction for the symmetric sample augmentation λ sweep. Values are mean $\pm$ standard deviation across seeds 35, 42, and 49.

λ	Final accuracy (%)	Final valid neuron fraction (%)
0.005	98.20 ± 0.09	0.01338 ± 0.00044
0.010	97.72 ± 0.12	0.01750 ± 0.00137
0.015	94.03 ± 0.61	0.08088 ± 0.02189
0.020	89.47 ± 0.76	0.11376 ± 0.00386
0.040	65.22 ± 26.78	0.15718 ± 0.13241

Table 6: Contrastive penalty λ/margin sweep. Each cell shows final valid neuron fraction (%). Values are mean $\pm$ standard deviation across seeds

λ	$m = -0.400$	$m = -0.200$	$m = 0.000$	$m = 0.200$	$m = 0.400$
0.005	0.01371 ± 0.00035	0.01400 ± 0.00037	0.01503 ± 0.00183	0.01464 ± 0.00084	0.01422 ± 0.00050
0.010	0.01418 ± 0.00022	0.01443 ± 0.00044	0.01654 ± 0.00364	0.01494 ± 0.00085	0.01398 ± 0.00054
0.040	0.01608 ± 0.00109	0.01505 ± 0.00064	0.02008 ± 0.01065	0.01374 ± 0.00135	0.01297 ± 0.00022
0.160	0.01504 ± 0.00200	0.01196 ± 0.00074	0.01184 ± 0.00075	0.01141 ± 0.00006	0.01050 ± 0.00061
0.640	0.01255 ± 0.00095	0.01014 ± 0.00095	0.00885 ± 0.00090	0.00802 ± 0.00044	0.00719 ± 0.00190

Table 7: Hard-freezing sweep. All runs start from the same trained core model.

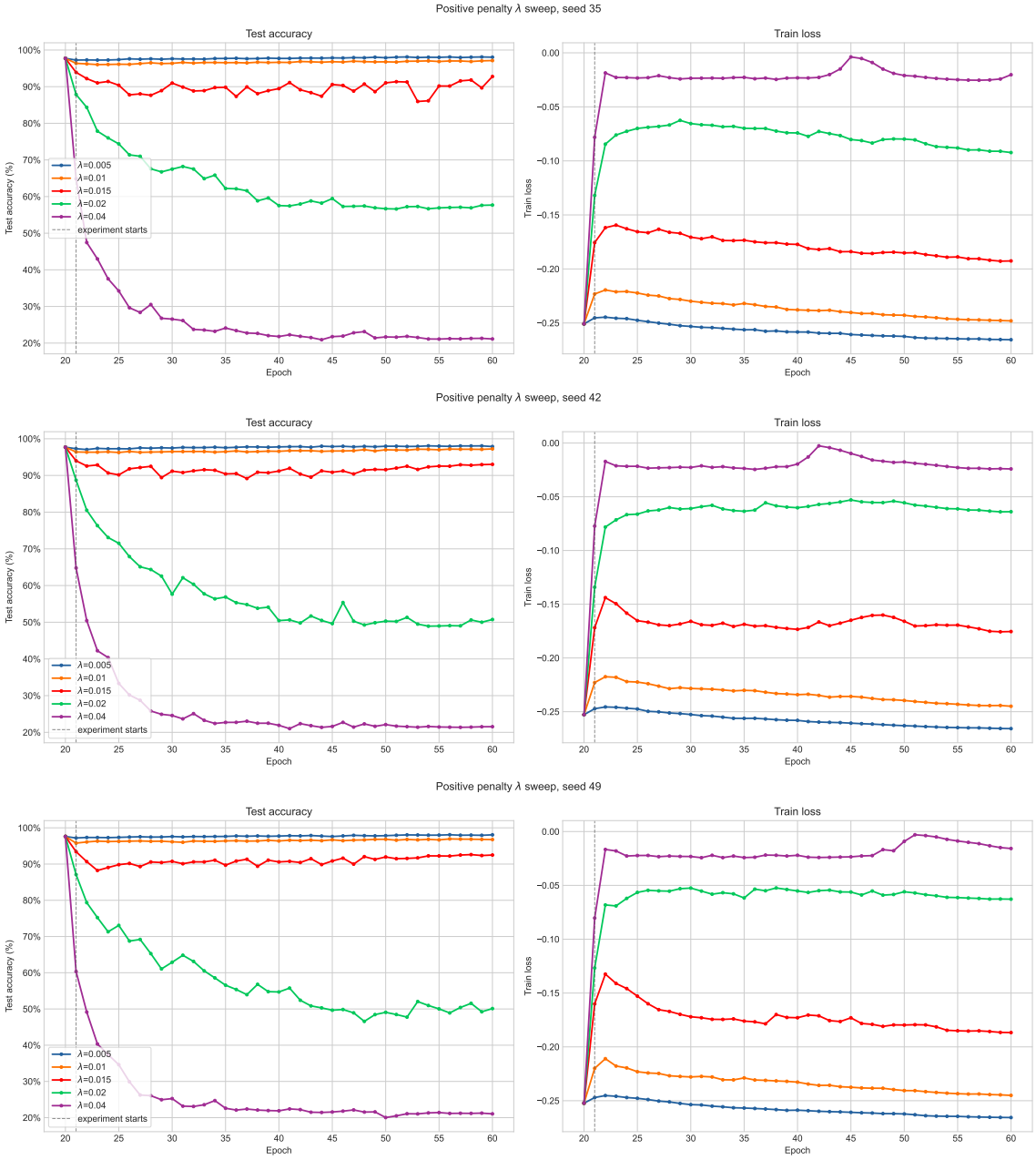
Frozen tail	Freeze fraction	Final accuracy (%)	Final valid neuron fraction (%)
Low ISI-CV	0.15	98.33	0.83513
Low ISI-CV	0.30	98.15	0.82620
Low ISI-CV	0.45	98.13	0.83207
Low ISI-CV	0.60	98.01	0.83967
High ISI-CV	0.15	97.57	0.85880
High ISI-CV	0.30	93.42	0.71333
High ISI-CV	0.45	92.46	0.63480
High ISI-CV	0.60	92.12	0.70553

Table 8: Valid neuron fraction for the soft-freezing p_{min} sweep. Values are mean $\pm$ standard deviation across seeds 35, 42, and 49.

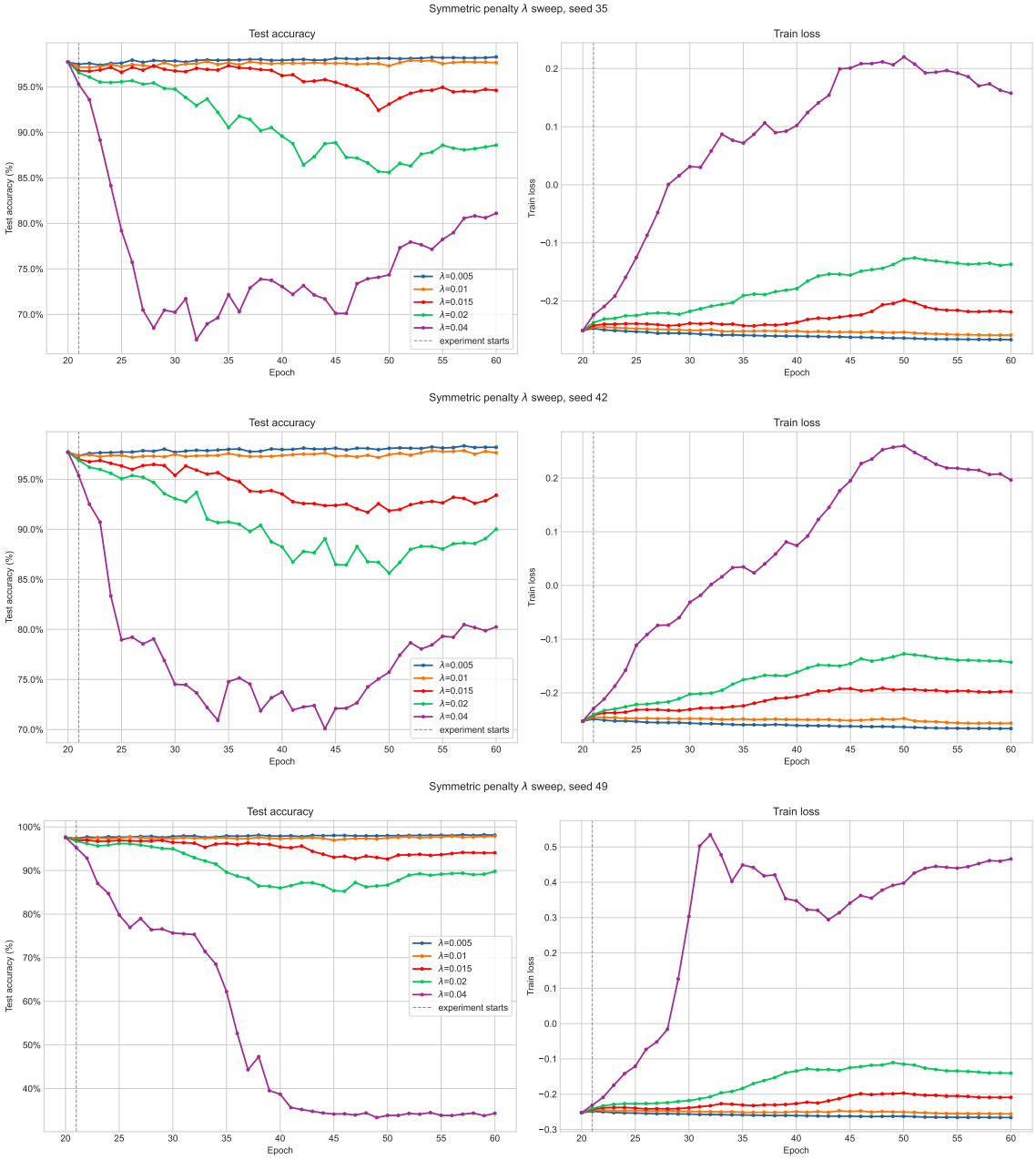
p_{min}	Final accuracy (%)	Final valid neuron fraction(%)
0.05	98.27 ± 0.04	0.87544 ± 0.04295
0.10	98.30 ± 0.12	0.87653 ± 0.04391
0.15	98.28 ± 0.16	0.88000 ± 0.04258
0.25	98.25 ± 0.15	0.87424 ± 0.03285
0.35	98.23 ± 0.14	0.88413 ± 0.03820
0.45	98.29 ± 0.06	0.88949 ± 0.03475
0.65	98.38 ± 0.10	0.88671 ± 0.02861
0.85	98.33 ± 0.04	0.87840 ± 0.02617

D Graphs

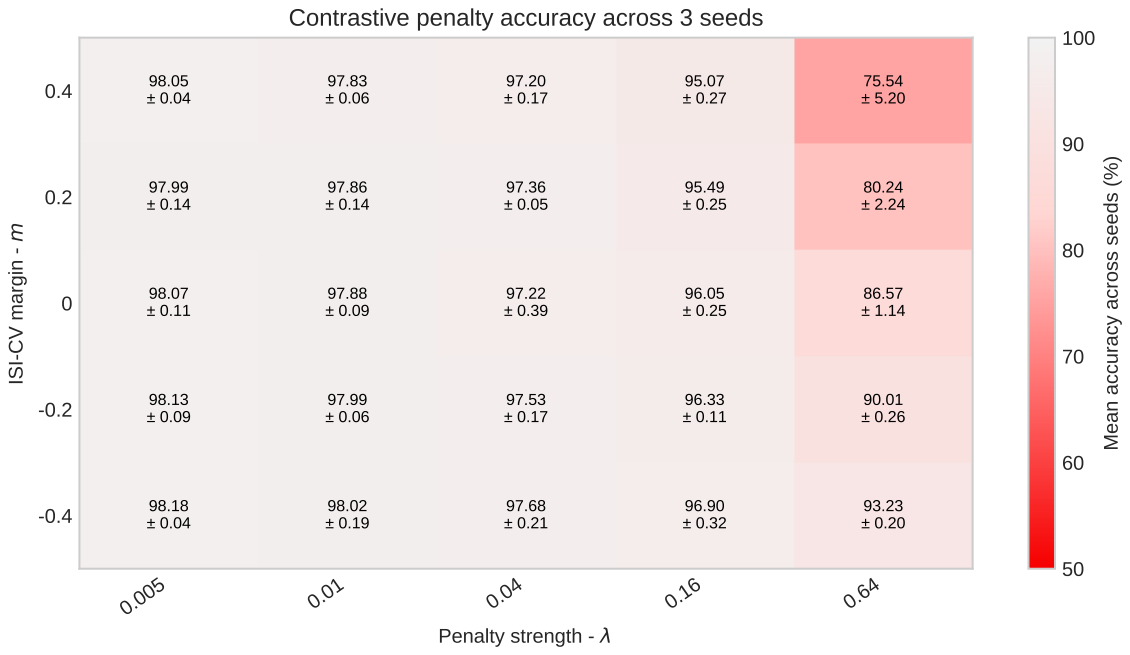
D.1 Penalty Positive

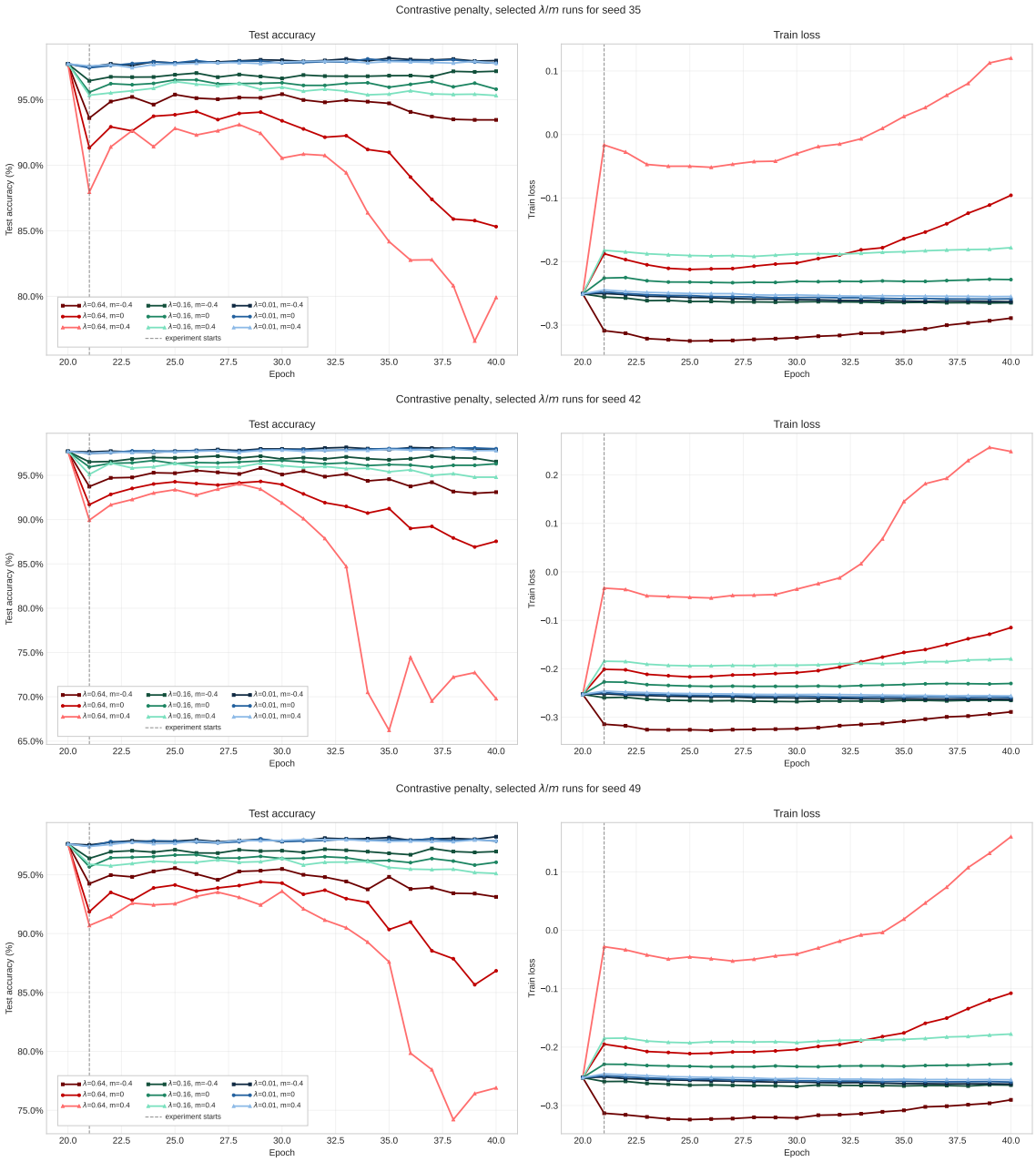


D.2 Penalty Symmetric

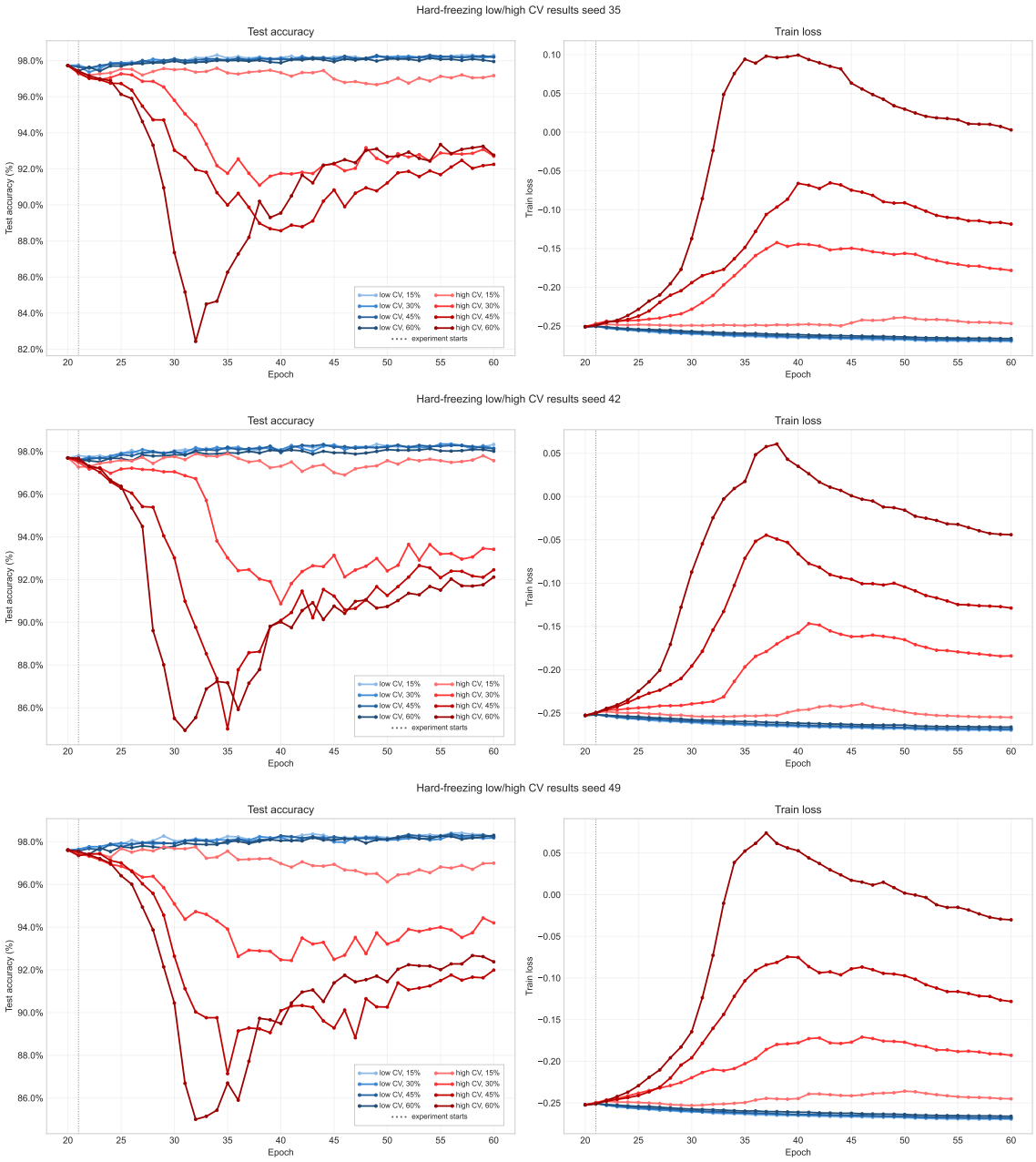


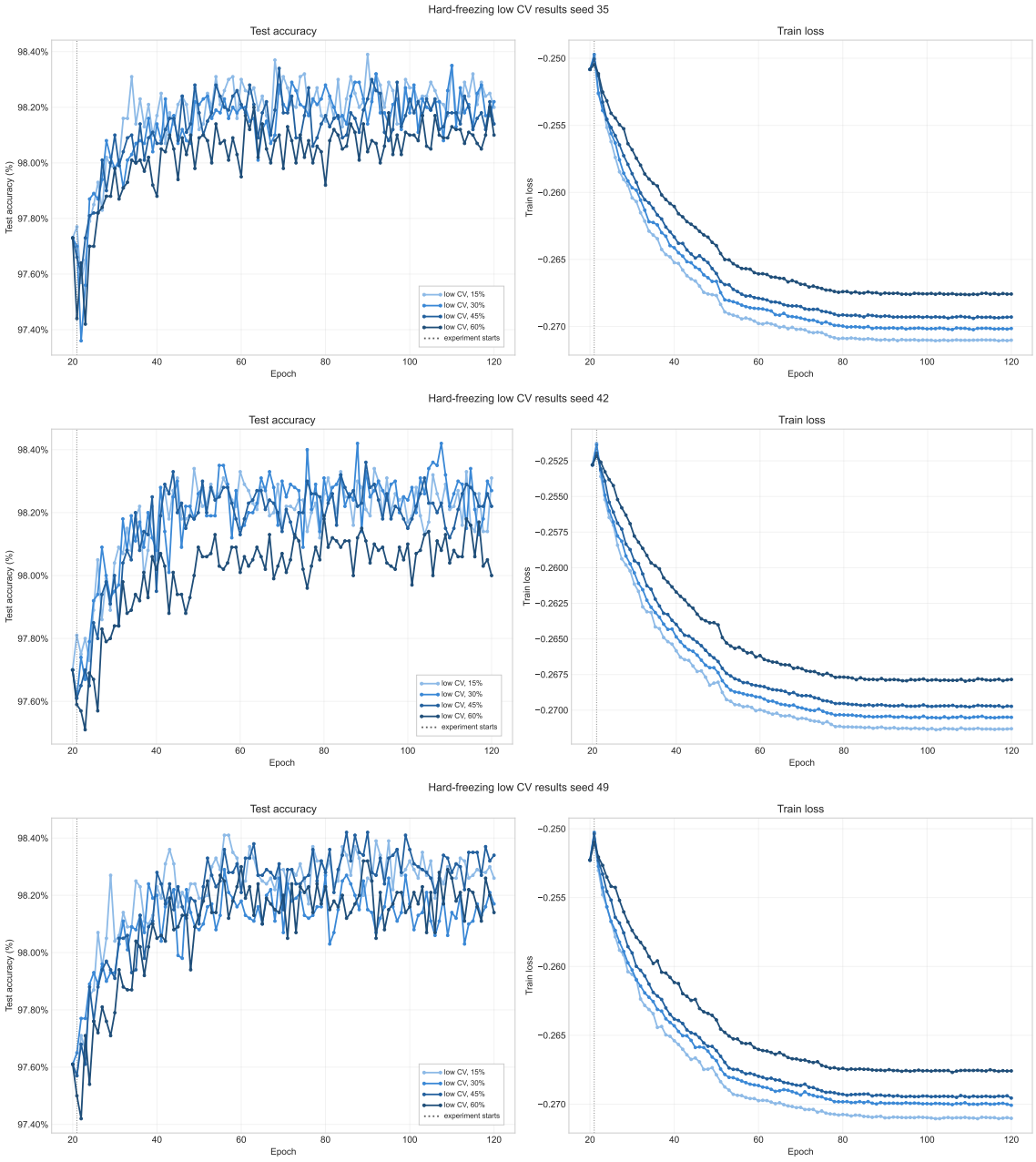
D.3 Contrastive Loss



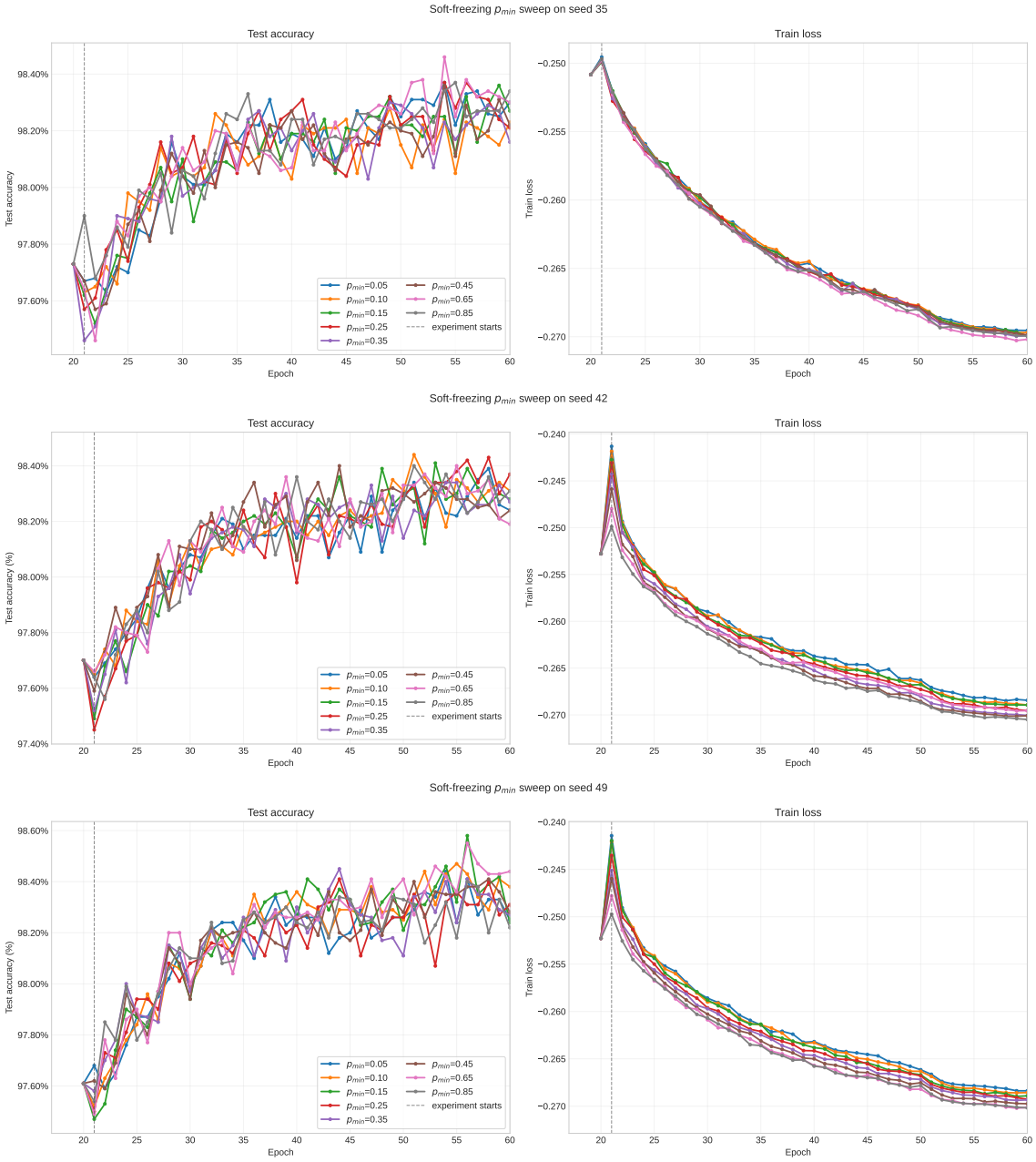

Contrastive penalty, selected λ/m runs for seed 35
Test accuracy
Test accuracy (%)
Epoch
Train loss
Train loss
Epoch
λ=0.64, m=-0.4
λ=0.64, m=0
λ=0.64, m=0.4
λ=0.16, m=-0.4
λ=0.16, m=0
λ=0.16, m=0.4
λ=0.01, m=-0.4
λ=0.01, m=0
λ=0.01, m=0.4
experiment starts
Contrastive penalty, selected λ/m runs for seed 42
Test accuracy
Test accuracy (%)
Epoch
Train loss
Train loss
Epoch
λ=0.64, m=-0.4
λ=0.64, m=0
λ=0.64, m=0.4
λ=0.16, m=-0.4
λ=0.16, m=0
λ=0.16, m=0.4
λ=0.01, m=-0.4
λ=0.01, m=0
λ=0.01, m=0.4
experiment starts
Contrastive penalty, selected λ/m runs for seed 49
Test accuracy
Test accuracy (%)
Epoch
Train loss
Train loss
Epoch
λ=0.64, m=-0.4
λ=0.64, m=0
λ=0.64, m=0.4
λ=0.16, m=-0.4
λ=0.16, m=0
λ=0.16, m=0.4
λ=0.01, m=-0.4
λ=0.01, m=0
λ=0.01, m=0.4
experiment starts

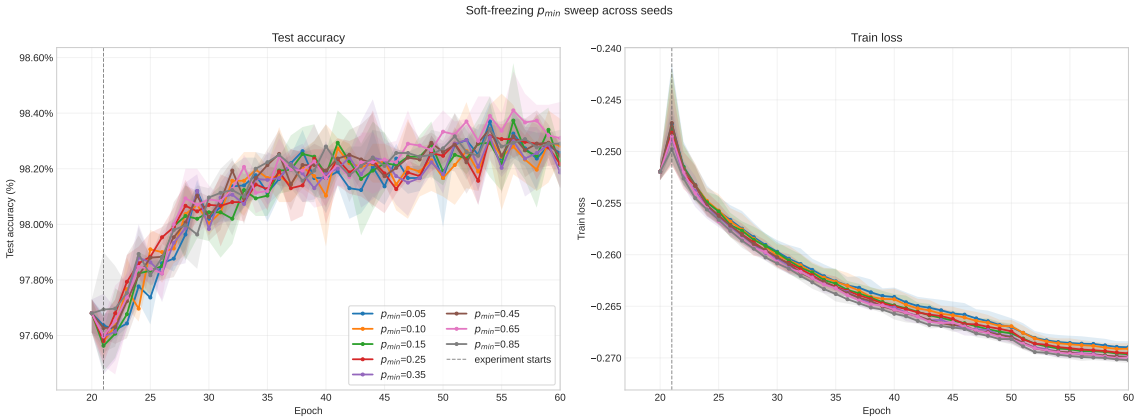
D.4 Hard Freezing



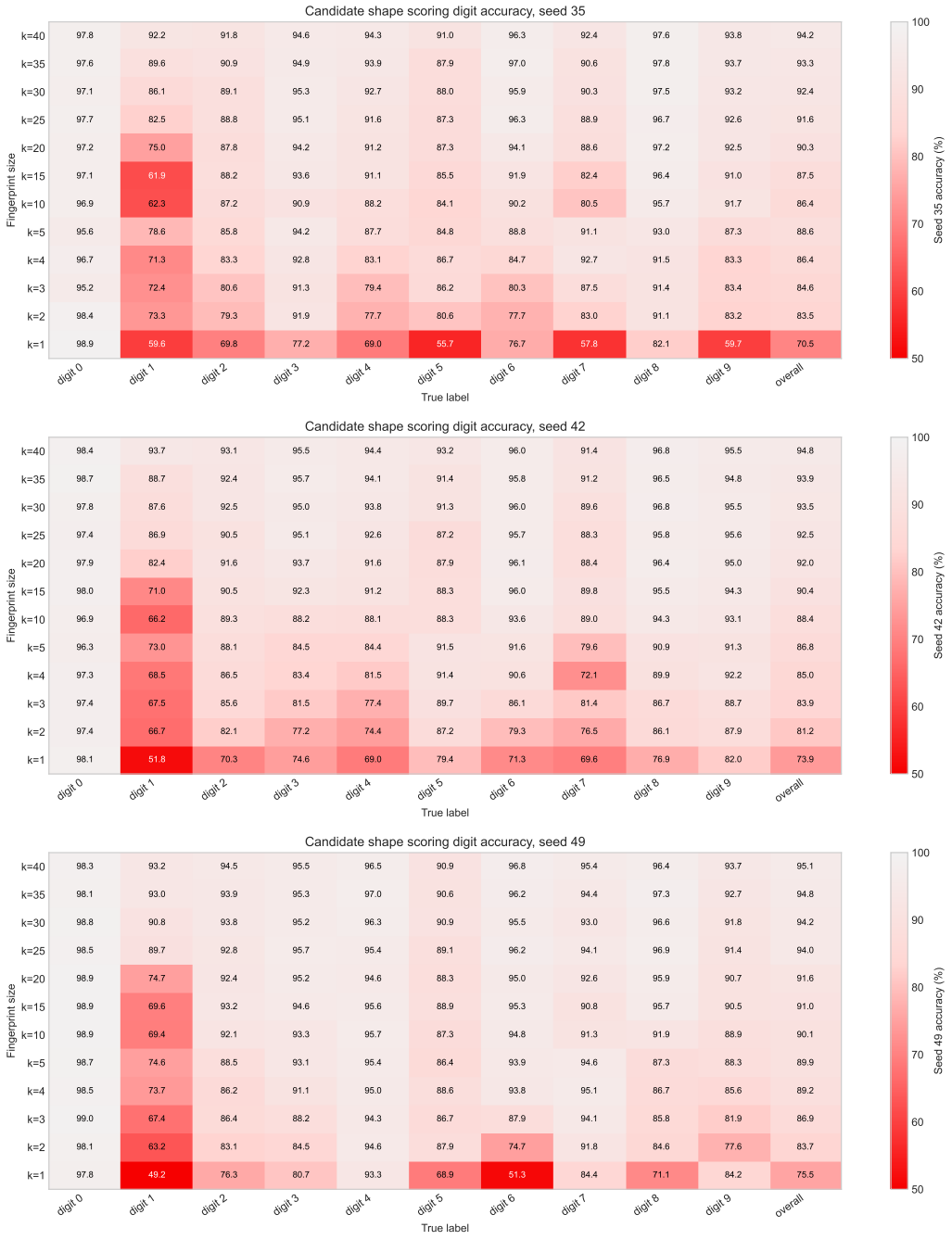

Hard-freezing low CV results seed 35
Test accuracy
Test accuracy (%)
98.40%
98.20%
98.00%
97.80%
97.60%
97.40%
20
40
60
80
100
120
Epoch
low CV, 15%
low CV, 30%
low CV, 45%
low CV, 60%
experiment starts
Train loss
Train loss
−0.250
−0.255
−0.260
−0.265
−0.270
Epoch
Hard-freezing low CV results seed 42
Test accuracy
Test accuracy (%)
98.40%
98.20%
98.00%
97.80%
97.60%
Epoch
Train loss
Train loss
−0.2525
−0.2550
−0.2575
−0.2600
−0.2625
−0.2650
−0.2675
−0.2700
Epoch
Hard-freezing low CV results seed 49
Test accuracy
Test accuracy (%)
98.40%
98.20%
98.00%
97.80%
97.60%
97.40%
Epoch
Train loss
Train loss
−0.250
−0.255
−0.260
−0.265
−0.270
Epoch

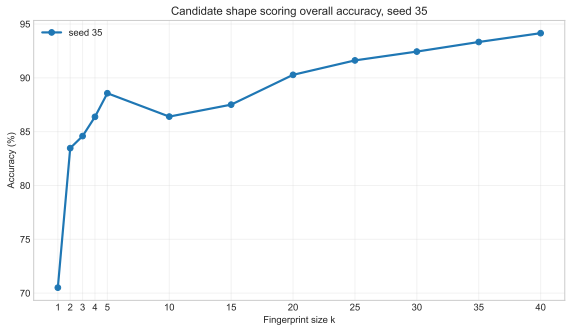
D.5 Soft Freezing

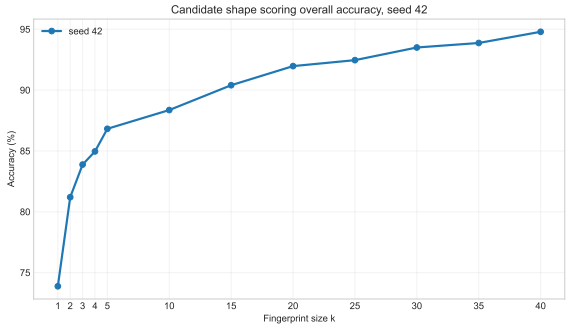


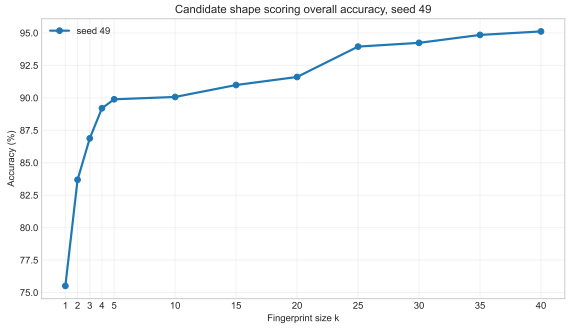

Soft-freezing p_{min} sweep across seeds
Test accuracy
Test accuracy (%)
98.60%
98.40%
98.20%
98.00%
97.80%
97.60%
20
25
30
35
40
45
50
55
60
Epoch
p_{min}=0.05
p_{min}=0.10
p_{min}=0.15
p_{min}=0.25
p_{min}=0.35
p_{min}=0.45
p_{min}=0.65
p_{min}=0.85
experiment starts
Train loss
Train loss
−0.240
−0.245
−0.250
−0.255
−0.260
−0.265
−0.270
20
25
30
35
40
45
50
55
60
Epoch

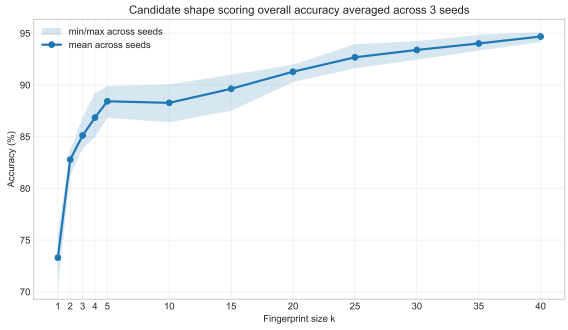
D.6 Candidate Scoring




Candidate shape scoring overall accuracy, seed 35
seed 35
Accuracy (%)
Fingerprint size k


Candidate shape scoring overall accuracy, seed 42
seed 42
Accuracy (%)
Fingerprint size k


Candidate shape scoring overall accuracy, seed 49
seed 49
Accuracy (%)
Fingerprint size k


Candidate shape scoring overall accuracy averaged across 3 seeds
min/max across seeds
mean across seeds
Accuracy (%)
Fingerprint size k

E Mathematical notations and formulas

Table 9: Notations used in the paper.

Symbol	Meaning
T	Number of simulation timesteps.
t	Timestep index.
i, j	Neuron indices, where j indexes a presynaptic (previous layer) neuron and i a postsynaptic (current layer) neuron.
N	Number of neurons in a layer.
x_i	Input pixel value of input neuron i.
x_{spike}	Poisson-encoded spiking input over T timesteps.
$u_i[t]$	Membrane potential of neuron i at timestep t.
$s_i[t]$	Binary spike emitted by neuron i at timestep t.
β	LIF membrane decay factor.
V_{th}	Firing threshold of the LIF neuron.
$w_{j,i}$	Synaptic weight from neuron j to neuron i.
r_i	Poisson firing probability associated with input pixel value x_i.
C_i	Spike count of neuron i over the full simulation window.
h	Layer spike-train activity over T timesteps.
h^{pos}	Layer activity for a positive input-label pair.
h^{neg}	Layer activity for a negative input-label pair.
$G(h)$	Standard Forward-Forward goodness computed from spike-count activity.
G^{pos}	Goodness of the standard FF positive pass.
G^{neg}	Goodness of the standard FF negative pass.
G_Δ	Standard FF goodness gap, $G^{pos} - G^{neg}$.
$\widetilde{G}^{pos}$	Modified positive goodness.
$\widetilde{G}^{neg}$	Modified negative goodness.
$\widetilde{G}_\Delta$	Modified goodness gap.
CV_{ISI}	Inter-spike interval coefficient of variation.
$\overline{CV}_{ISI}^{pos}$	Mean ISI-CV computed from positive-sample spike trains.
$\overline{CV}_{ISI}^{neg}$	Mean ISI-CV computed from negative-sample spike trains.
$CV_{ISI\Delta}$	ISI-CV gap, defined as $\overline{CV}_{ISI}^{neg} - \overline{CV}_{ISI}^{pos}$.
λ	Weighting coefficient controlling the strength of an ISI-CV term.
ρ	Scaling factor used to match the magnitude of the ISI-CV.
m	Margin used in the contrastive gap loss.
$\mathcal{L}$	Total training loss.
$\mathcal{L}_{FF}$	Standard Forward-Forward loss.
$\mathcal{L}_{CV}$	Contrastive ISI-CV loss term.
α	Sharpness parameter of the SiLU loss.
η	Learning rate.
F	Set of neurons selected for hard freezing.
z_i	Binary frozen-neuron mask for neuron i, where $z_i = 0$ freezes updates and $z_i = 1$ keeps updates active.
f_i	Soft-freezing plasticity factor assigned to neuron i.
$p_{\min}$	Minimum plasticity factor used in soft freezing.
v_i	ISI-CV score used to rank neuron i during soft freezing.
$\text{rank}(v_i)$	Rank of neuron i according to v_i within the selected soft-freezing group.
n	Number of neurons included in the soft-freezing ranking.
A	Number of reference samples used to compute class fingerprints.
K	Fingerprint size, i.e. number of selected neurons per class.
F_q	Fingerprint neuron set selected for class q.
$cs(x, q)$	Candidate score for input x under candidate label q.
$\hat{y}(x)$	Predicted label for input x.