Practical Bayesian Inference for Speech SNNs: Uncertainty and Loss-Landscape Smoothing

Yesmine Abdennadher
Idiap Research Institute
Martigny, Switzerland
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
yesmine.abdennadher@idiap.ch

Philip N. Garner
Idiap Research Institute
Martigny, Switzerland
phil.garner@idiap.ch

Abstract

Spiking Neural Networks (SNNs) are naturally suited for speech processing tasks due to their specific dynamics, which allows them to handle temporal data. However, the threshold-based generation of spikes in SNNs intuitively causes an angular or irregular predictive landscape. We explore the effect of using the Bayesian learning approach for the weights on the irregular predictive landscape. For the surrogate-gradient SNNs, we also explore the application of the *Improved Variational Online Newton* (IVON) approach, which is an efficient variational approach. The performance of the proposed approach is evaluated on the Heidelberg Digits and Speech Commands datasets. The hypothesis is that the Bayesian approach will result in a smoother and more regular predictive landscape, given the angular nature of the deterministic predictive landscape. The experimental evaluation of the proposed approach shows improved performance on the negative log-likelihood and Brier score. Furthermore, the proposed approach has resulted in a smoother and more regular predictive landscape compared to the deterministic approach, based on the one-dimensional slices of the weight space.

CCS Concepts

• **Computing methodologies** → **Machine learning**; **Artificial intelligence**; • **Mathematics of computing** → **Probability and statistics**.

Keywords

Spiking neural networks, Bayesian learning, IVON, uncertainty, Speech recognition

ACM Reference Format:
Yesmine Abdennadher and Philip N. Garner. 2026. Practical Bayesian Inference for Speech SNNs: Uncertainty and Loss-Landscape Smoothing. In *International Conference on Neuromorphic Systems (ICONS 2026), August 04–06, 2026, Chicago, IL, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3822454.3822477

1 Introduction

Spiking Neural Networks are well suited for speech processing because they are fundamentally temporal in nature. In principle, they can also achieve low-latency and energy-efficient "event-driven" inference. Unlike traditional neural networks, in SNNs the communication between neurons is in the form of spikes, and the state is maintained using the dynamics of the neurons.

Nevertheless, training SNNs is not without difficulties. In practice, surrogate gradients enable the optimization of SNNs, where the derivative of the spike function is replaced by a smooth function during backpropagation. However, the forward pass is based on threshold-based spike generation, and this intuitively leads to an irregular optimization landscape. Specifically, this gives rise to the following question: Do deterministic SNNs result in a less regular predictive objective?

If this is true, then relying on a single deterministic estimate may not be optimal. In this case, a single estimate may not be informative about the weights and may not result in a highly robust predictive objective, especially when this is locally irregular. Therefore, it is important to look beyond deterministic training and consider distributions over weights.

In this work, we study *Improved Variational Online Newton* (IVON) as a practical Bayesian learning method for surrogate-gradient SNNs. In simple terms, IVON replaces a single deterministic set of weights with a learned Gaussian distribution over weights. It updates both the posterior mean and uncertainty-related statistics during minibatch training, while remaining compatible with the same surrogate-gradient SNN training pipeline. This makes IVON a scalable way to introduce Bayesian weight uncertainty without requiring a separate posterior-fitting procedure or an ensemble of models. This is particularly appealing in the context of SNNs because learning under weight uncertainty can be interpreted as optimizing a locally averaged predictive objective, which may smooth irregular behavior induced by threshold-based spike dynamics.

This leads to the central hypothesis of our work: if the local predictive landscape of a deterministic SNN is angular or piecewise irregular due to spike-threshold switching, then Bayesian learning over the weights may be especially beneficial. In particular, averaging predictions over a distribution of plausible weights may yield a smoother and less brittle predictive objective than deterministic point-estimate training.

We apply IVON on the Heidelberg Digits (HD) and Speech Commands (SC) benchmarks using predictive and uncertainty metrics, as well as one-dimensional weight-space slices. Our intention is to verify if the quality of the predictions is improved by the application of IVON and if the predictive landscape is smoother and more regular than in the case of deterministic training.

Contributions. We make three contributions: we examine IVON as an effective solution to weight uncertainty learning in surrogate



This work is licensed under a Creative Commons Attribution 4.0 International License.
ICONS 2026, Chicago, IL, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2808-2/2026/08
https://doi.org/10.1145/3822454.3822477

gradient SNNs for speech recognition; we propose that IVON in SNNs is equivalent to regularizing the objective function of prediction by means of weight averaging in Bayesian methods; we provide supporting evidence in predictive, calibration, and local loss geometry analyses.

2 Related work

Bayesian methods aim to improve predictive reliability by modeling uncertainty over neural network parameters. Many approaches include mean-field variational inference such as Bayes by Backprop [3, 8], dropout-based approximations [7, 11], stochastic-gradient MCMC methods [19, 21], and Gaussian approximations derived from curvature information or optimization trajectories [6, 12, 15]. Among post-hoc methods (i.e., applied after training), Laplace approximation is a prominent approach [6]. In practice, however, scalability often motivates fitting it only on the last layer. In SNNs, this places uncertainty on the readout weights while keeping the spiking feature extractor fixed, and therefore does not capture perturbations that modify the internal spiking dynamics.

Variational methods are particularly attractive because they maintain a weight distribution while remaining compatible with standard minibatch training [9, 10, 14]. IVON [16] belongs to this family and can be viewed as a Bayesian learning rule in which both parameter means and uncertainty are updated jointly during training. It was introduced as a more practical variant of earlier online variational Newton methods, with reported empirical improvements over related approaches.

In parallel, SNNs have been successfully applied to speech and audio tasks with surrogate-gradient training [2, 20], but Bayesian learning in SNNs remains relatively underexplored. Prior work has mainly adapted variational Bayesian learning rules to spiking models [17]. This question is also practically relevant beyond training: in neuromorphic SNN deployments, effective parameter perturbations can arise from device mismatch, fabrication variability, quantization, and other hardware non-idealities, and these effects have been shown to degrade SNN inference and motivate robustness-oriented training [1, 4, 5]. In threshold-driven networks such as SNNs, even small parameter changes can alter spike timings or spike occurrences and thereby affect predictions. Our focus is therefore narrower: we study IVON specifically because its training-time weight perturbations are directly aligned with our central question of whether learning under weight uncertainty can yield a smoother and less perturbation-sensitive predictive objective.

3 Task and data description

We consider the problem of speech recognition, in which we wish to classify each short spoken utterance into one of a fixed number of classes. We are interested in two of the most commonly used benchmarks in previous work on this problem: *Heidelberg Digits (HD)* and *Speech Commands (SC)*. In both datasets, we begin with a raw audio waveform and compute a set of acoustic features, which are then input to the model as a sequence of feature vectors over a fixed window of time.

The HD dataset has spoken digits in English and German, with 20 classes, recorded from twelve different speakers, with two speakers only in the test set. The training set has 8,332 examples, the test set has 2,088 examples, and no validation set. The SC dataset uses Google Speech Commands v0.2, which has 35 keyword classes drawn from a diverse set of speakers. Using the standard split, the training set has 75,466 examples, the validation set has 9,981 examples, and the test set has 20,382 examples [18].

Although spiking versions of these datasets are also available (e.g., spike-based event representations specifically suited for training SNNs), we observed that in our pipeline, the feature-based (non-spiking) versions of these datasets produce stronger and more stable results for our speech recognition baselines [2]. We thus present our main results on HD and SC using the feature-based inputs and a unified evaluation protocol for both deterministic and Bayesian models.

4 SNN Architecture and Training

We use the speech input data to train the model, which is based on the feedforward spiking neural network (SNN) with leaky integrate and fire (LIF) neurons.

The structure of the network is composed of two hidden layers of LIF neurons and a spike-rate readout head. At any discrete point in time, denoted as $t \in \{1, \dots, T\}$, layer ℓ is provided with an input vector $x_t^{(\ell)}$ and applies a linear transformation followed by LIF membrane integration to produce a spike output. Each of the two hidden layers of LIF neurons is composed of a fully connected synaptic layer, batch normalization (BN) to ensure stable training, and dropout regularization with a given drop probability p_{drop}. This is our proposed deterministic model, and it is also utilized as a backbone in our SNN models in conjunction with IVON in the Bayesian setting.

For the discrete-time LIF dynamics: Let $u_t^{(\ell)}$ denote the membrane potential and $s_t^{(\ell)} \in \{0,1\}^{n_\ell}$ the spike vector at time t in layer ℓ. A standard discrete-time update is

$$u_{t+1}^{(\ell)} = \alpha^{(\ell)} u_t^{(\ell)} + W^{(\ell)} x_t^{(\ell)} - v_{\text{th}}\, s_t^{(\ell)}, \tag{1}$$

$$s_t^{(\ell)} = \mathbb{I}\Big[u_t^{(\ell)} \geq v_{\text{th}}\Big], \tag{2}$$

where $\alpha^{(\ell)} \in (0,1)$ controls the leak, $W^{(\ell)}$ are trainable synaptic weights, and v_{th} is a firing threshold. Since the spike function is non-differentiable, training relies on surrogate gradients, here using a boxcar surrogate function [13], while keeping the forward dynamics strictly spiking.

For classification, we use a spike rate readout, which transforms the spike train of the last hidden layer into a single utterance-level prediction. We denote the spike vector of the last hidden layer at time step t as $s_t^{(L)}$. To obtain the average firing rate over the fixed window of T time steps, we use the average firing rate

$$r = \frac{1}{T} \sum_{t=1}^{T} s_t^{(L)}. \tag{3}$$

The resulting rate vector is then mapped to class logits via a linear classifier,

$$z = W_{\text{out}}\, r + b_{\text{out}}. \tag{4}$$

In our implementation, the readout head follows the same stabilization pattern as the hidden blocks: we apply batch normalization

and dropout to the rate features before the final linear layer. Importantly, the spike-rate readout aggregates the last-layer spike train over the fixed window and produces class logits, which are required both for the cross-entropy training objective and for computing calibrated predictive probabilities (via softmax) and uncertainty metrics.

4.1 Angularity of the deterministic loss landscape

The key issue that this work addresses is whether the prediction goal of surrogate gradient SNNs is locally smooth with respect to perturbations of the network parameters. In regular neural networks, changes to the weights of the network cause changes to the output of the network that are also smooth with respect to changes to the loss of the network. However, for SNNs, the forward pass of the network also involves the generation of spikes, and this generation of spikes can change based on the perturbation of the weights of the network.

In order to study this phenomenon, we study slices of the deterministic cross entropy loss function near a trained parameter vector. Let $\hat{\theta}$ be the learned parameter vector, and d be a unit-norm direction in parameter space. We introduce

$$\mathcal{L}(\alpha) = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \mathrm{CE}\big(y, f_{\hat{\theta}+\alpha d}(x)\big), \tag{5}$$

where $\mathcal{B}$ is a fixed set of minibatches and α controls the perturbation magnitude.

The resulting curve in figure 1 has an angular or piecewise irregular form, rather than a smooth form. The loss curve shows abrupt changes in slope and sharp transitions, which is consistent with the underlying model, as the spiking process naturally leads to such effects, even if the surrogate gradients have a smoothing effect on the backpass, as the forward process still depends on the threshold crossings. Neurons can abruptly begin or cease firing, or change the timing of their spikes, and such abrupt changes propagate through the network to the readout.

This gives us direct evidence that the deterministic local loss landscape in SNNs is locally non-smooth and highly sensitive to small parameter perturbations.

5 Practical Bayesian Learning Method

The following sections are tailored to test the hypothesis that the application of Bayesian learning can mitigate the locally angular predictive behavior of deterministic SNNs. The rationale for this hypothesis is based on the observation that the prediction in SNNs with surrogate gradients ultimately depends on the threshold-based event of spikes. Therefore, small changes in the weights can cause the neurons to fire or not to fire, which in turn can propagate over time and cause abrupt changes in the predictive objective. Therefore, the locally angular predictive behavior of deterministic SNNs might remain. In this context, the application of Bayesian learning is of special interest because it replaces the point estimate of the weights with a distribution over the weights and the prediction with the average over the perturbations in the weights. The application of this approach is expected to mitigate the locally angular predictive behavior of SNNs.

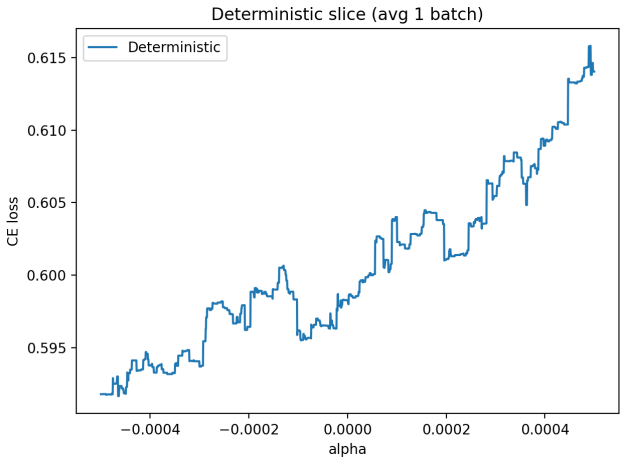


Figure 1: Local one-dimensional slice of the deterministic cross-entropy loss on a fixed batch.

A practical approach to Bayesian deep learning is to approximate the intractable posterior distribution over the weights with a tractable family of distributions, whose parameters are optimized during training. A common choice is a mean-field Gaussian approximation, which remains simple enough for minibatch optimization while still providing uncertainty estimates over the model parameters. In this work, we adopt this variational perspective and focus on *Improved Variational Online Newton* (IVON) [16] as a practical method for optimizing such an approximate posterior in surrogate-gradient SNNs. We first introduce the mean-field formulation and then describe the IVON updates used in our experiments.

5.1 Mean-Field Variational Inference

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ denote the training set and θ the network parameters. Since the exact posterior $p(\theta \mid \mathcal{D})$ is generally intractable, variational inference introduces a tractable approximation $q(\theta)$ and optimizes it to be close to the true posterior. In the mean-field setting, the parameters are assumed independent under $q(\theta)$, leading to a diagonal Gaussian approximation

$$q(\theta) = \mathcal{N}(\mu, \mathrm{diag}(\sigma^2)), \tag{6}$$

where μ and σ are learned variational parameters.

Rather than learning a single point estimate of the weights, variational Bayesian learning seeks a distribution over plausible parameters by maximizing the evidence lower bound (ELBO),

$$\mathcal{L}_{\mathrm{ELBO}}(\mu, \sigma) = \mathbb{E}_{q(\theta)}\left[\sum_{i=1}^{N} \log p(y_i \mid x_i, \theta)\right] - \mathrm{KL}(q(\theta) \,\|\, p(\theta)), \tag{7}$$

or equivalently by minimizing the negative ELBO. The first term encourages good predictive fit to the data, while the Kullback–Leibler term regularizes the approximate posterior toward the prior $p(\theta)$.

The expectation with respect to $q(\theta)$ is estimated by Monte Carlo sampling. Using the reparameterization trick, a sample from the

approximate posterior can be written as

$$\theta^{(s)} = \mu + \sigma \odot \varepsilon^{(s)}, \qquad \varepsilon^{(s)} \sim \mathcal{N}(0, I), \tag{8}$$

which gives the estimator

$$\mathbb{E}_{q(\theta)}\left[\sum_{i=1}^{N} \log p(y_i \mid x_i, \theta)\right] \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N} \log p\left(y_i \mid x_i, \theta^{(s)}\right). \tag{9}$$

In practice, the sum over the dataset is replaced by a minibatch estimate and the variational parameters are updated by stochastic optimization.

With an isotropic Gaussian prior $p(\theta) = \mathcal{N}(0, \lambda^{-1}I)$, the KL term has the closed form

$$\mathrm{KL}(q(\theta) \,\|\, p(\theta)) = \frac{1}{2}\sum_{j}\left[\lambda\left(\mu_j^2 + \sigma_j^2\right) - \log\left(\lambda\sigma_j^2\right) - 1\right]. \tag{10}$$

5.2 IVON: Improved Variational Online Newton

IVON [16] is a scalable variational method that bridges Bayesian learning and conventional minibatch training. IVON uses a Gaussian approximation for the weights as follows:

$$q_{\mathrm{IVON}}(\theta) = \mathcal{N}(\mu, \Sigma), \tag{11}$$

where μ is used for representing means for the parameters, and Σ is for representing weight uncertainty. This is important in our context since we are able to incorporate Bayesian learning into SNN training using surrogate gradient without modifying the SNN structure.

In the training process, IVON draws samples of the weights from the current approximate posterior distribution, calculates the gradients of the minibatch, and performs updates on the posterior mean and some statistics concerning the curvature that controls the variance of the approximate posterior distribution. Unlike the first-order mean-field variational inference, in which the variance in each direction of the parameters cannot be adjusted based on the curvature information, IVON makes use of those statistics to make adjustments in the variance of each direction of the parameters according to the constraint from the data.

The curvature information used by IVON comes from the Newton-style view of optimization. Around the current parameter value, the minibatch training loss can be locally approximated by a second-order Taylor expansion,

$$\bar{\ell}(\theta) \approx \bar{\ell}(\theta_t) + g_t^\top(\theta - \theta_t) + \frac{1}{2}(\theta - \theta_t)^\top H_t(\theta - \theta_t), \tag{12}$$

where $g_t = \nabla_\theta \bar{\ell}(\theta_t)$ is the minibatch gradient and $H_t = \nabla_\theta^2 \bar{\ell}(\theta_t)$ is the Hessian of the minibatch loss. The Hessian describes the local curvature of the loss surface. Therefore, it provides information about how sensitive the loss is to changes in each parameter direction, which is useful for determining the uncertainty of the approximate posterior.

The calculation of the entire Hessian matrix H_t is computationally impractical in neural networks. IVON does not use the entire Hessian matrix but rather approximates the curvature using a stochastic diagonal version of the Hessian. Through the Gaussian reparameterization, we have a sample parameter vector represented by

$$\theta_t = m_t + \sigma_t \odot \xi_t, \qquad \xi_t \sim \mathcal{N}(0, I), \tag{13}$$

where m_t is the current posterior mean, σ_t is the posterior standard deviation, ξ_t is a standard Gaussian perturbation, and $\odot$ denotes element-wise multiplication. Equivalently, if $\epsilon_t = \theta_t - m_t$, then ϵ_t/σ_t is the normalized perturbation used to generate the sampled weights.

This reformulated sampling method enables IVON to approximate the diagonal curvature from the first-order minibatch gradient as well as the sample perturbation. More specifically, the Hessian diagonal elements could be approximated element-wise as

$$\mathrm{diag}(H_t) \approx \hat{g}(\theta_t) \odot \frac{\epsilon_t}{\sigma_t}, \tag{14}$$

where $\hat{g}(\theta_t) = \nabla_\theta \bar{\ell}(\theta_t)$ is the minibatch gradient evaluated at the sampled weights. Thus, the reparameterization trick is not used to compute the full Hessian directly; instead, it makes it possible to obtain an efficient stochastic diagonal curvature estimate from sampled weights and ordinary gradients. This avoids explicit second-order differentiation, Hessian storage, and Hessian inversion.

As a consequence, the ability of IVON to recompute the posterior mean and the posterior uncertainties during the same minibatch training process makes it possible to train IVON along with the standard SNN model in the usual training loop. The extra overhead in the computational effort involved in the training process is due to maintaining the posterior statistics rather than an ensemble of models, Markov chain Monte Carlo, or any other posterior fitting method.

IVON is normally used with a Gaussian prior over the weights, which corresponds to ℓ_2-type regularization with standard scaling. IVON also incorporates an effective sample size, which is represented by ESS. This balances the relative importance of the likelihood and prior depending on the data set size. In our experiments, we used ESS = N, where N is the number of training data.

At test time, predictions are made via Bayesian model averaging. For a given x^*, S weight configurations are sampled from the learned posterior and the averaged predictive probabilities are used:

$$p(y^* \mid x^*, \mathcal{D}) \approx \frac{1}{S}\sum_{s=1}^{S} p(y^* \mid x^*, \theta^{(s)}), \qquad \theta^{(s)} \sim q_{\mathrm{IVON}}(\theta). \tag{15}$$

6 Experimental evaluation of IVON

All models are trained for 30 epochs. In both deterministic and Bayesian settings, we use a learning-rate scheduler that reduces the step size when validation performance plateaus.

For the deterministic baseline, we use Adam with learning rate 10^{-3}, batch size 128, and dropout $p_{\mathrm{drop}} = 0.1$.

For the Bayesian setting, we keep the same network architecture and replace Adam with IVON. On Heidelberg Digits (HD), IVON uses learning rate 0.1, batch size 16, dropout $p_{\mathrm{drop}} = 0$, and weight decay 10^{-5}. On Speech Commands (SC), we use the same learning rate and weight decay, with batch size 32 and dropout $p_{\mathrm{drop}} = 0$. We additionally freeze the batch-normalization running statistics in the Bayesian model, which we found empirically to be more stable under sampled weight perturbations. We set the effective sample size to ESS = N, where N is the number of training examples.

For prediction at test time, Bayesian predictions are calculated through the Monte Carlo averaging over (S=20) weight configurations sampled. The value of the number of Monte Carlo samples (S) is an inference-time hyperparameter as opposed to being a necessity in the process. Using (S=20) is a trade-off made between the quality of the predictions and computation costs. When the samples of the forward pass are evaluated serially, the inference time is directly proportional to (S), hence (S=20) can mean a (20×) multiplication of the inference cost of a forward pass without sampling. When less samples are needed, it will depend on the importance of the latency; otherwise, it can be higher for more consistent results. As the samples are independent, they can also be executed in batches/parallel when possible.

6.1 Evaluation Metrics

We will use the test of predictive quality and uncertainty quality on the validation/test split, using appropriate measures of classification quality, predictive uncertainty, and calibration.

Negative Log-Likelihood (NLL).. For the predicted probabilities $p_\theta(y \mid x)$, the NLL is given by

$$\mathrm{NLL} = -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(y_i \mid x_i). \tag{16}$$

Unlike accuracy, NLL checks for both correctness and confidence.

Brier score. In addition, we provide the multiclass Brier score, which estimates the squared difference between the predicted probabilities and the corresponding one-hot encoded target probabilities. Let $p_\theta(y = k|x_i)$ be the predicted probabilities for each class k given the input x_i. Let $\mathbb{I}[y_i = k]$ be the corresponding one-hot encoded target probabilities. The multiclass Brier score is given by

$$\mathrm{Brier} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{C} \left(p_\theta(y{=}k \mid x_i) - \mathbb{I}[y_i = k]\right)^2. \tag{17}$$

Low values indicate good probabilistic predictions, capturing both accuracy and calibration.

Predictive entropy. We use the entropy of the predictive distribution to quantify predictive uncertainty:

$$\mathcal{H}\big[p(y \mid x)\big] = -\sum_{k=1}^{C} p(y{=}k \mid x)\, \log p(y{=}k \mid x). \tag{18}$$

Expected entropy and mutual information. We use S Monte Carlo samples of the latent variable, denoted by $\{\theta^{(s)}\}_{s=1}^{S}$ and probabilities $p^{(s)}(y \mid x) = p(y \mid x, \theta^{(s)})$. We compute the expected entropy

$$\mathbb{E}\big[\mathcal{H}\big[p(y \mid x, \theta)\big]\big] \approx \frac{1}{S}\sum_{s=1}^{S} \mathcal{H}\big[p^{(s)}(y \mid x)\big], \tag{19}$$

and the predictive mutual information;

$$\mathcal{I}\left(y, \theta \mid x\right) = \mathcal{H}[\mathbb{E}_\theta p(y \mid x, \theta)] - \mathbb{E}_\theta\big[\mathcal{H}\big[p(y \mid x, \theta)\big]\big], \tag{20}$$

where mutual information measures epistemic uncertainty.

Calibration and expected calibration error (ECE).. We measure the calibration of our model, which means that our confidence should match our actual correctness. We used the expected calibration error (ECE) with confidence bins B to measure the calibration. For each data point, let $\hat{p}_i = \max_k p(y{=}k \mid x_i)$ and $\hat{y}_i = \arg\max_k p(y{=}k \mid x_i)$. Then

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{I}_b|}{n} \left|\mathrm{acc}(\mathcal{I}_b) - \mathrm{conf}(\mathcal{I}_b)\right|, \tag{21}$$

where $\mathcal{I}_b$ is the set of examples in bin b.

6.2 Results

Table 1 shows that IVON improves predictive performance over the deterministic baseline on both datasets. On Heidelberg Digits (HD), where the deterministic model is already very strong, IVON increases accuracy from 98.10% to 99.51% and substantially reduces both NLL (from 0.0553 to 0.0233) and Brier score (from 0.0294 to 0.0105). The entropy is also lower, indicating more concentrated predictions on average, while ECE remains at a similarly low level. This suggests that, even in a near-saturated regime, IVON improves the quality of the predictive distribution beyond the gain in accuracy alone.

The effect is more pronounced on Speech Commands (SC), which is the more challenging task. Here, IVON improves accuracy from 77.01% to 79.93%, while also reducing NLL, ECE, entropy, and Brier score. In particular, the drop in ECE from 0.0224 to 0.0133 indicates better calibration, and the reduction in Brier score from 0.3184 to 0.2838 shows that the predictive probabilities are better aligned with the observed outcomes. The nonzero mutual information (MI) obtained with IVON on both datasets further indicates the presence of a measurable epistemic uncertainty component, which is absent in the deterministic point estimate.

Overall, these results suggest that IVON provides benefits beyond raw classification accuracy. It consistently yields better probabilistic predictions, as reflected by lower NLL and Brier score on both benchmarks, with the largest overall gains on SC. This is consistent with the view that learning under weight uncertainty can improve both predictive reliability and uncertainty quality in surrogate-gradient SNNs.

6.3 Loss-surface smoothness via 1D weight-space slices

The angularity discussion in Section 4.1 motivates investigating whether Bayesian prediction decreases local sensitivity to parameter perturbations. To this end, we assess one-dimensional slices of the loss around a learned solution. To make the slice analysis reproducible and less arbitrary, we choose the perturbation direction as the normalized gradient direction at the trained solution. Specifically, let $\hat{\theta}$ be the learned parameter vector and let $\mathcal{B}$ be a fixed set of minibatches. We compute

$$d = \frac{\nabla_\theta \mathcal{L}(\hat{\theta}; \mathcal{B})}{\left|\nabla_\theta \mathcal{L}(\hat{\theta}; \mathcal{B})\right|_2}, \tag{22}$$

where d is the direction of steepest first-order change in the loss. We then evaluate the loss along $\hat{\theta} + \alpha d$ for a range of perturbation

Dataset	Method	Acc(%) ↑	NLL ↓	ECE ↓	Entropy ↓	MI ↑	Brier ↓
HD	Deterministic (MAP)	98.10	0.0553	0.0097	0.0824	–	0.0294
	IVON (MC S=20)	99.51	0.0233	0.0103	0.0450	0.0163	0.0105
SC	Deterministic (MAP)	77.01	0.8063	0.0224	0.7967	–	0.3184
	IVON (MC S=20)	79.93	0.7191	0.0133	0.6829	0.0711	0.2838

Table 1: Test-set results on Heidelberg Digits (HD) and Speech Commands (SC).

values α. Although this remains a one-dimensional probe of a high-dimensional loss surface, it provides a controlled way to compare how deterministic and Bayesian predictive losses respond to the same local parameter perturbation.

For the deterministic model, the slice is defined as

$$\mathcal{L}(\alpha) = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \mathrm{CE}\left(y, f_{\hat{\theta}+\alpha d}(x)\right), \tag{23}$$

where α controls the perturbation magnitude.

For IVON, we assess the Bayesian predictive loss by averaging the predictive probabilities over sampled weight configurations from the learned posterior,

$$\mathcal{L}_{\mathrm{Bayes}}(\alpha) = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \left[-\log\left(\frac{1}{S} \sum_{s=1}^{S} p\left(y \mid x, \theta^{(s)}(\alpha)\right)\right)\right], \tag{24}$$
$$\theta^{(s)}(\alpha) \sim q(\theta;\alpha).$$

where $q(\theta;\alpha)$ denotes the IVON posterior centered on the perturbed mean $\hat{\theta} + \alpha d$ along the same slice direction.

Figure 2 shows the local one-dimensional slice of the IVON Bayesian predictive loss around the learned solution, evaluated on a fixed batch. As the perturbation parameter α varies along the gradient direction, the Bayesian predictive loss changes in a relatively gradual and regular manner. Although the curve is not perfectly quadratic and still exhibits mild local fluctuations, it is visibly less jagged and less step-like than its deterministic counterpart. This suggests that posterior averaging in IVON attenuates sharp threshold-induced variations in the local predictive objective. In other words, averaging predictions over a distribution of plausible weights softens the effect of abrupt spike-pattern changes, yielding a smoother and less brittle local loss landscape.

Improvements in predictive accuracy, especially in negative log-likelihood, Brier score, and uncertainty-based measures, suggest that Bayesian approaches can provide better predictive distributions than deterministic SNN models. In addition, the smoother behavior of the curves and the finite-difference smoothness measures are consistent with the idea that accounting for weight uncertainty can alleviate local irregularities due to threshold-based spike generation.

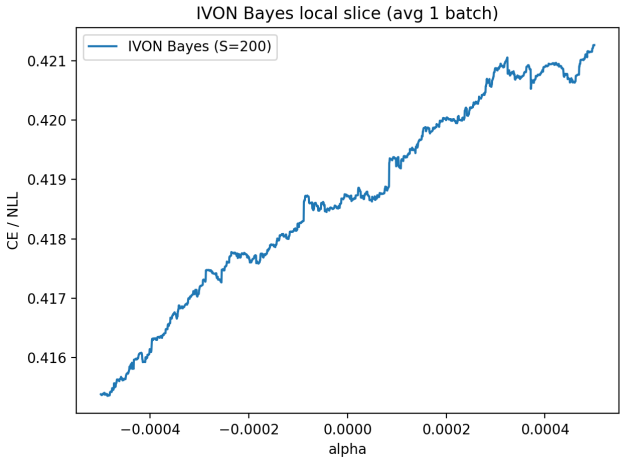


Figure 2: Local one-dimensional slice of the IVON Bayesian predictive loss on a fixed batch.

7 Discussion

It can be seen that our findings show the possibility of using IVON to incorporate weight uncertainty in surrogate gradient-based SNNs. From a computational perspective, IVON remains compatible with the standard minibatch training pipeline. It does not require exact Bayesian inference, full Hessian computation, MCMC sampling, or ensemble training. Instead, it maintains a Gaussian approximation to the weight posterior using feasible diagonal curvature statistics that are updated together with the network parameters. The additional computation comes from posterior sampling and the maintenance of uncertainty-related statistics, while preserving a single-model training procedure. This provides a practical trade-off: IVON introduces Bayesian weight uncertainty within the usual training loop, while avoiding substantially heavier posterior-inference alternatives such as ensembles or sampling-based Bayesian methods.

While our experiments have been conducted using speech recognition, the IVON-Bayesian SNN framework is not restricted to speech datasets. In fact, IVON manipulates the weight distribution of the SNN model itself, without any assumptions about input features of the speech data. Thus, the very same technique can be used for other SNN-related problems where predictions depend on uncertainties in the weight distributions of the network. This could include temporal or event-based domains, such as neuromorphic vision, event cameras, sensors, time series and many others. The key factor is that the SNN can be trained using surrogate gradient descent.

It is important to note that this comparison with the deterministic SNN baseline was deliberately chosen because it offers the closest point-estimate approximation of IVON in terms of the architecture, the data pipeline, and the surrogate gradient-based training procedure used. By doing so, we are able to observe how substituting a fixed set of trained weights for the distribution over weights influences the predictions of the network. As this research focuses

on examining whether Bayesian weight uncertainty can help increase the robustness of the predictive ability of the model, this controlled experiment gives an answer to this question. Further comparisons of our approach with other similar SNN techniques are left for future research.

8 Conclusion

Combined, the findings indicate that the Bayesian consideration of the uncertainty around the SNN weights has the effect of reducing the local irregularity resulting from thresholded spikes. This is because weight perturbation in deterministic SNNs, due to thresholding, tends to generate locally irregular objectives to predict. On the other hand, Bayesian models take into account the weight uncertainty, which may have the effect of smoothing out the irregularity. In line with this understanding, the Bayesian model performed significantly better at providing good predictive distributions compared to the corresponding deterministic model. In particular, it scored lower in negative log-likelihood, Brier scores, among other measures of uncertainty.

Acknowledgments

We acknowledge the support of the Swiss National Science Foundation under grant number 223717 (MORPhyN).

References

[1] Abhronil Bhattacharjee et al. 2022. Examining the Robustness of Spiking Neural Networks on Non-Ideal Memristive Crossbars. In *GLSVLSI*.

[2] Alexandre Bittar and Philip N. Garner. 2022. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience* Volume 16 - 2022 (2022). doi:10.3389/fnins.2022.865897

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Networks. In *International Conference on Machine Learning (ICML)*.

[4] Julian Büchel, Dmitrii Zendrikov, Sergio Solinas, Giacomo Indiveri, and Dylan R. Muir. 2021. Supervised training of spiking neural networks for robust deployment on mixed-signal neuromorphic processors. *Scientific Reports* (2021).

[5] Uğurcan Çakal, Maryada, Chenxi Wu, Ilkay Ulusoy, and Dylan R. Muir. 2023. Gradient-descent hardware-aware training and deployment for mixed-signal neuromorphic processors. *arXiv preprint arXiv:2303.12167* (2023).

[6] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. 2021. Laplace Redux - Effortless Bayesian Deep Learning. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 20089–20103. https://proceedings.neurips.cc/paper_files/paper/2021/file/a7c9585703d275249f30a088cebba0ad-Paper.pdf

[7] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1050–1059. https://proceedings.mlr.press/v48/gal16.html

[8] Alex Graves. 2011. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (Eds.), Vol. 24. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf

[9] Mohammad Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. 2018. Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 2611–2620. https://proceedings.mlr.press/v80/khan18a.html

[10] Mohammad Emtiyaz Khan and Håvard Rue. 2023. The Bayesian learning rule. *J. Mach. Learn. Res.* 24, 1, Article 281 (Jan. 2023), 46 pages.

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. https://arxiv.org/abs/1412.6980

[12] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A Simple Baseline for Bayesian Uncertainty in Deep Learning. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/118921efba23fc329e6560b27861f0c2-Paper.pdf

[13] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine* 36 (2019), 51–63. https://api.semanticscholar.org/CorpusID:207941413

[14] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. 2019. Practical Deep Learning with Bayesian Principles. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/b53477c2821c1bf0da5d40e57b870d35-Paper.pdf

[15] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. A Scalable Laplace Approximation for Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Skdvd2xAZ

[16] Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Clement Bazan, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. 2024. Variational learning is effective for large deep networks. JMLR.org.

[17] Nicolas Skatchkovsky, Hyeryung Jang, and Osvaldo Simeone. 2022. Bayesian continual learning via spiking neural networks. *Frontiers in Computational Neuroscience* Volume 16 - 2022 (2022). doi:10.3389/fncom.2022.1037976

[18] Pete Warden. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. (04 2018). doi:10.48550/arXiv.1804.03209

[19] Max Welling and Yee Whye Teh. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Bellevue, Washington, USA) *(ICML'11)*. Omnipress, Madison, WI, USA, 681–688.

[20] Jibin Wu, Emre Yılmaz, Malu Zhang, Haizhou Li, and Kay Chen Tan. 2020. Deep Spiking Neural Networks for Large Vocabulary Automatic Speech Recognition. *Frontiers in Neuroscience* Volume 14 - 2020 (2020). doi:10.3389/fnins.2020.00199

[21] Difan Zou, Pan Xu, and Quanquan Gu. 2019. Stochastic Gradient Hamiltonian Monte Carlo Methods with Recursive Variance Reduction. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.