CERTAINTY-GUIDED FEATURE SEPARATION FOR EARLY DECISION IN SPIKING KEYWORD SPOTTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Keyword spotting (KWS) requires low latency, low energy consumption, and high recognition accuracy, making efficient early decision-making particularly important for resource-constrained edge devices. Existing early-decision methods for spiking neural networks (SNNs) mainly supervise cumulative outputs across time steps, while feature-level approaches often enforce uniform temporal separation and overlook the progressively increasing discriminative information in speech. In this work, we propose a certainty-guided spiking feature separation framework that dynamically estimates the informativeness of each time step from the SNN's predictive distribution and uses it to weight intermediate feature separation. By applying a certainty-guided feature separation objective combined with a dynamic training schedule, unreliable separation are suppressed during early training and discriminative representation learning are strengthened as the classifier becomes more reliable. Experiments on GSC V2 and MSWC Microset datasets demonstrate that our method improves early decision accuracy while maintaining or improving final recognition accuracy, achieving better performance across multiple time steps. The proposed framework provides a lightweight and inference-free approach to learning temporally adaptive spiking representations, offering a promising direction for low-latency and energy-efficient KWS on edge devices.

1 INTRODUCTION

Keyword spotting (KWS) is a fundamental capability for smartphones, smart speakers, in-vehicle audio systems, and various edge speech-interaction devices, where response latency, energy consumption, and recognition accuracy are central design requirements López-Espejo et al. (2021); Zhang et al. (2017); Choi et al. (2019). Spiking neural networks (SNNs) communicate through discrete spike events and are attractive for event-driven, energy-efficient temporal processing Maass (1997); Davies et al. (2018); Neftci et al. (2019). Recent SNN-based speech and KWS systems have demonstrated the suitability of spiking dynamics for acoustic sequence modeling Yılmaz et al. (2020); Yang et al. (2022); Song et al. (2024b); Bittar & Garner (2022); Wang et al. (2024); Yin et al. (2021); Bellec et al. (2018). However, many of these systems classify a complete input segment rather than explicitly optimizing predictions at every observed prefix Yılmaz et al. (2020); Yang et al. (2022); Song et al. (2024b); Bittar & Garner (2022); Wang et al. (2024). Consequently, they do not fully exploit the ability of SNNs to update their states online, leaving room for better latency-accuracy trade-offs.

To address this issue, ED-sKWS Song et al. (2024a) introduces an early-decision framework. By applying cross-entropy supervision to accumulated outputs at every time step, it improves intermediate predictions and enables confidence-based early stopping. However, accurate early decisions require the network to form discriminative feature representations as early as possible: the earlier class-separable features emerge, the earlier reliable predictions can be made. In ED-sKWS, intermediate features receive only indirect supervision through the output layer. It does not explicitly organize the feature space into class-discriminative clusters at early time steps, which may limit the reliability of early predictions.

Contrastive learning provides a direct way to shape the discriminative structure of learned representations by attracting positive pairs and separating negative pairs Chen et al. (2020); Khosla et al. (2020). Siamese temporal contrastive learning (STCL) extends this principle to SNNs by construct-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

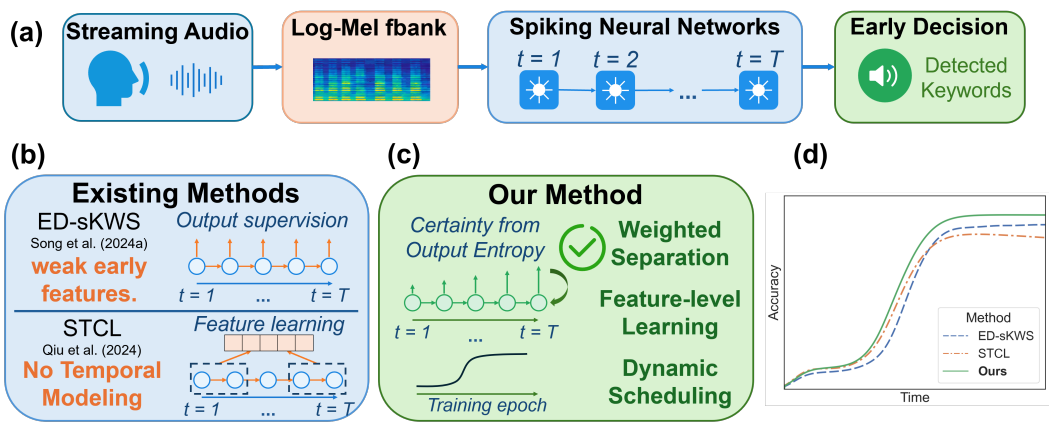


Figure 1: Overview of our framework. (a) Early-decision keyword spotting task. Given an input speech signal, the goal is to recognize keywords as early as possible using an SNN. (b) Existing methods only introduce supervision at the output level, or use identical early-late features at the feature level. (c) Our method introduces weighted and separated feature-level learning and dynamic scheduling. (d) Our method achieves higher accuracy than existing methods at both early and late stages.

ing positives across time steps and, in its supervised form, across samples sharing a class label Qiu et al. (2024). Although this strategy improves low-latency SNN classification, applying uniform temporal alignment to early-decision KWS can be problematic because the available speech evidence changes across time. Aggressively enforcing separation or alignment at early time steps may reinforce representations formed from insufficient evidence. This raises two questions: (1) how can the model estimate whether the current spiking state contains useful class-discriminative evidence, and (2) how should it regulate feature separation across time and training progress?

We propose a certainty-guided spiking feature separation learning framework, as shown in Figure 1. The key idea is to exploit the predictive distribution produced by the classification readout of the SNN at each time step to estimate the certainty of the class information encoded in the current spiking state. This certainty is then used as a time-step-wise weight for the feature separation loss, encouraging the network to form class-separable representations earlier at reliable time steps, thereby accelerating early decisions while preserving accuracy. Specifically, we introduce a certainty estimator based on predictive entropy and a certainty-weighted feature separation loss that simultaneously pulls intra-class representations together and pushes inter-class representations apart. To address the unreliability of predictive uncertainty during the early stages of SNN training, we further introduce a dynamic weight scheduling strategy that enables coarse-to-fine optimization of the spiking representation space. Importantly, the proposed mechanism is only introduced during training and removed during inference, requiring no additional computational operations and thus incurring no extra latency or energy overhead.

The main contributions are summarized as follows:

- We propose an early-decision learning framework, enabling reliable class-discriminative capability at an earlier stage without requiring additional manual annotations of temporal decision boundaries.
- We use the SNN's own predictive certainty as a weighting signal to guide the separation of intermediate spiking features and further introduce a dynamic weight scheduling strategy for spiking representation learning.
- Experiments on the GSC V2 and MSWC Microset datasets demonstrate that the proposed framework improves the accuracy of early decisions while preserving the accuracy of late decisions.

2 Related Works

2.1 Early Decision for Sequence Classification

The core challenge of online sequence classification is deciding when the observed prefix supports a sufficiently accurate prediction, thereby balancing earliness against error Xing et al. (2009); Schäfer & Leser (2020); Renault et al. (2024). This temporal-prefix problem differs from layer-wise early exit and adaptive computation, which reduce network depth or recurrent computation after the input is available Teerapittayanon et al. (2016); Graves (2016). In speech recognition, HuBERT-EE adds self-attention exit branches to intermediate HuBERT layers and trains the branches with connectionist temporal classification objectives Yoon et al. (2022); Hsu et al. (2021). DAISY uses entropy derived from self-supervised intermediate predictions to select an exit without task-specific branch retraining Lin et al. (2024). Splitformer places exit decoders throughout a Conformer encoder and supervises their intermediate CTC predictions Lasbordes et al. (2025); Gulati et al. (2020). These methods show that explicit intermediate supervision and sample-dependent stopping can reduce computation, although their layer-wise exit setting is distinct from deciding after a partial speech sequence.

For spiking keyword spotting, convolutional, residual, adaptive-recurrent, and global–local SNNs exploit event-driven computation for efficient acoustic processing Yılmaz et al. (2020); Yang et al. (2022); Bittar & Garner (2022); Wang et al. (2024). Spiking-LEAF further adapts a learnable audio frontend to spike-based speech processing Zeghidour et al. (2021); Song et al. (2024b). Most of these systems target the prediction from a complete segment. ED-sKWS instead introduces a CT loss that supervises cumulative outputs at every time step and combines it with confidence-based early stopping Song et al. (2024a). Temporal efficient training also supervises every time step, but scores each step independently and therefore does not accumulate the evidence observed so far Deng et al. (2022). These results establish the feasibility of temporal early decisions in SNN-based KWS and their potential to reduce latency and energy use.

2.2 Recognition of Uncertain Information in Sequential Data

When observations arrive sequentially, a decision rule must determine whether the evidence seen so far is reliable enough to act on. Predictive confidence and certainty are common proxies for this purpose, but modern neural-network probabilities can be miscalibrated, especially under distribution shift Guo et al. (2017); Ovadia et al. (2019). Bayesian dropout, deep ensembles, and decompositions of aleatoric and epistemic uncertainty provide richer alternatives, albeit with additional modeling or inference cost Gal & Ghahramani (2016); Lakshminarayanan et al. (2017); Kendall & Gal (2017). Among lightweight early-exit methods, BranchyNet uses branch confidence, while DeeBERT and FastBERT use predictive entropy or related uncertainty scores Teerapittayanon et al. (2016); Xin et al. (2020); Liu et al. (2020). PABEE instead requires prediction consistency across consecutive exits Zhou et al. (2020), and CALM dynamically allocates computation during autoregressive generation according to intermediate confidence Schuster et al. (2022). In early time-series classification, ECTS estimates a minimum reliable prefix under a nearest-neighbor formulation, whereas TEASER learns a second-stage classifier that decides whether a prefix prediction is trustworthy Xing et al. (2009); Schäfer & Leser (2020). Together, these studies motivate reliability-aware decisions while also showing that a confidence score should be treated as a proxy rather than a guarantee of correctness.

2.3 Feature Representation Learning in SNNs

Feature-level objectives provide supervision beyond the final classifier. Supervised contrastive learning attracts same-class representations while separating representations from different classes Khosla et al. (2020). STCL extend this idea across SNN time steps: temporal views of the same example and representations sharing a class label form positives, while the remaining pairs provide contrastive negatives Qiu et al. (2024). Auxiliary learning has also been used to regularize SNN features with additional training tasks Cachi et al. (2023), and detachable accumulation pathways provide dense auxiliary signals while retaining a fully spiking inference model Chen et al. (2023). More generally, deep supervision and feature hints demonstrate that intermediate objectives can improve the discriminativeness and trainability of hidden representations Lee et al. (2015);

Romero et al. (2014). Enhanced self-distillation for SNNs maps firing-rate representations through lightweight artificial-neural-network branches and explicitly filters unreliable self-generated knowledge Zhao et al. (2026). These works establish the utility of training-only auxiliary supervision, but they do not adapt the strength of feature separation to the reliability of each temporal prefix.

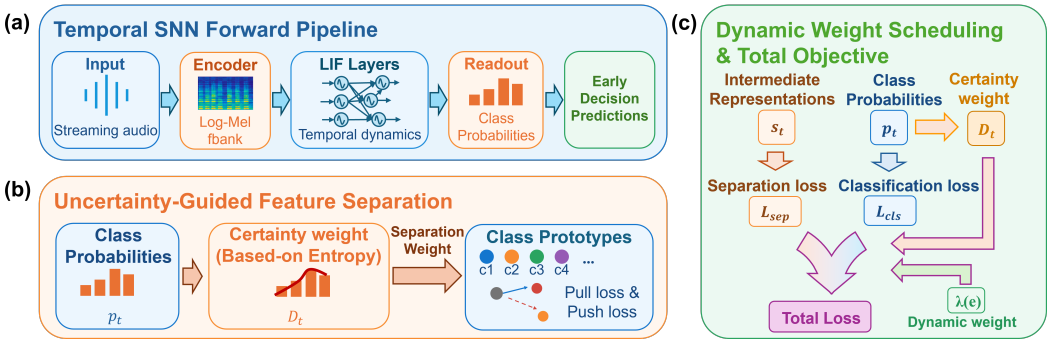


Figure 2: Our proposed spiking keyword spotting method. (a) Spiking keyword spotting pipeline. (b) Certainty is computed from the entropy of the network output distribution and used as the weighting for the feature separation strength. (c) The separation loss and classification loss on the intermediate representation are weighted by certainty and dynamically scheduled over epochs during training to obtain the total loss.

3 Methods

3.1 Overall Architecture

Let a speech sequence be represented as a temporal sequence of acoustic features $X = (x_1, x_2, \ldots, x_T)$, where T denotes the length of the complete sequence. In real-time speech interaction scenarios, the system can only observe the prefix available up to the current time step, denoted by $X_t = (x_1, \ldots, x_t)$. It is required to produce a prediction of class probabilities $p_t(y) = P(y \mid X_t)$ at each time step t, resulting in a continuous early classification process over the speech sequence $p_1, p_2, \ldots, p_T$, as shown in Figure 2(a).

The acoustic input is first converted by a spiking encoder into a temporal sequence $X = (x_1, x_2, \ldots, x_T)$, where $x_t \in \mathbb{R}^d$ denotes the acoustic features at the t-th time step. The sequence is then fed into the SNN, which produces a prediction at every time step while continuously updating its internal spiking representation as additional acoustic information becomes available. Specifically, the spike output generated by the leaky integrate-and-fire (LIF) layers at time step t is denoted as s_t. As the intermediate spiking representation, s_t encodes the accumulated temporal information of the input up to the current time step. The final layer transforms s_t into a probability distribution p_t over the keyword classes. The network is trained using the standard surrogate-gradient approach for SNNs, which approximates the non-differentiable spiking function with a differentiable surrogate during backpropagation.

3.2 Feature Separation Loss

Building upon the cross-entropy classification loss L_{cls}, to facilitate early decision-making, the representation at each time step should be close to the prototype of its target class while remaining sufficiently separated from prototypes of other classes. We therefore impose a feature separation loss on the spiking outputs of intermediate SNN layers, as shown in Figure 2(b). For each class y, we maintain a learnable class prototype c_y. Given the hidden outputs s_t, its squared Euclidean distance to the target class prototype is defined as

$$d(s_t, c_y) = \|s_t - c_y\|^2.$$

Our feature separation loss combines center-based intra-class attraction with a contrastive hinge penalty against non-target prototypes. It consists of

$$L_{sep}(s_t) = d(s_t, c_y) + \alpha \sum_{j \neq y} \max\big(0,\, m - d(s_t, c_j)\big),$$

where the first term Pull Loss attracts the sample representation toward its ground-truth class prototype, and the second term Push Loss enforces a hinge margin m between the sample and prototypes of other classes.

We apply the separation objective to the intermediate spiking representations throughout the entire temporal sequence. The resulting sequence-level feature separation loss is

$$L_{sep}^{all} = \frac{1}{T} \sum_{t=1}^{T} L_{sep}(s_t).$$

This objective encourages the SNN to progressively organize its temporal representations into class-discriminative clusters, providing a basis for reliable early decisions.

3.3 CERTAINTY-GUIDED SEPARATION STRENGTH

Applying the feature separation loss with equal strength at all time steps can force the network to cluster spiking patterns before sufficient information has been observed, potentially reinforcing erroneous representations and contaminating the class prototypes. We therefore introduce the predictive certainty inherent in the SNN's own outputs to estimate whether the current spiking state contains sufficient class-discriminative information, and to dynamically control the feature separation strength at each time step accordingly.

Given the class probability distribution at time step t, $p_t = [p_t^1, \ldots, p_t^C]$, we first compute its predictive entropy Shannon (1948) $H(p_t) = -\sum_i p_t^i \log p_t^i$, and then normalize the entropy by the logarithm of the number of classes, which is used for the early-exit confidence score in FastBERT Liu et al. (2020):

$$U_t = \frac{H(p_t)}{\log C} \in [0, 1].$$

A value of $U_t \to 1$ indicates that the predictive distribution is close to uniform, suggesting that the current observation provides insufficient discriminative information. In contrast, $U_t \to 0$ indicates that one class dominates the predictive distribution, suggesting that the current observation contains more informative evidence. We therefore define the predictive certainty D_t as

$$D_t = 1 - U_t \in [0, 1],$$

in which D_t is higher when the predictive distribution is more concentrated and confident.

We use this certainty measure to weight the feature separation loss at each time step:

$$L_{sep}^{weighted} = \frac{1}{T} \sum_{t=1}^{T} D_t L_{sep}(s_t).$$

In this way, the network's own predictive reliability determines when feature separation should be enforced, enabling self-guided representation learning along the temporal dimension.

3.4 DYNAMIC WEIGHT SCHEDULING ALONG THE LEARNING PROCESS

However, predictive confidence is also a imperfect proxy: the output probabilities can be miscalibrated and become less reliable under distribution shift Guo et al. (2017); Ovadia et al. (2019). The

effectiveness of D_t as a measure of information certainty therefore relies on an important prerequisite: the network must already possess a basic ability to classify speech, such that $p_t(y \mid X_t; \theta)$ meaningfully reflects the class information contained in X_t. Especially during the early stages of training, the predictive distribution p_t is often close to uniform. In this regime, a low D_t reflects the network's own lack of predictive capability rather than a genuine lack of discriminative information in the current speech segment.

As training progresses, the main-task supervision L_{cls} gradually establishes the basic mapping from spikes to predictions, $s_t \to p_t$. Accordingly, D_t gradually transitions from reflecting **network-level certainty** to reflecting **input-information certainty** Kendall & Gal (2017). Therefore, the reliability of D_t must be established progressively during training.This motivates scheduling the contribution of L_{sep} over the course of training: its influence is suppressed during early training and gradually increased as the main task becomes more stable.

We introduce a training-stage-dependent coefficient $\lambda(e)$, where e denotes the current training epoch, and formulate the overall objective as

$$L(e) = (1 - \lambda(e))L_{\text{cls}} + \lambda(e)L_{\text{sep}}^{weighted}.$$

Here, L_{cls} establishes the basic speech classification capability and stable spiking response patterns, while L_{sep} further organizes the spiking representation space to enhance early class separability. We require $\lambda(e)$ to satisfy two properties: $\lambda(e) \approx 0$ at the beginning of training, so that feature separation is disabled or strongly suppressed; and $\lambda(e) \to \lambda_{\max}$ as training progresses, allowing the increasingly reliable D_t to fully guide representation learning.

Inspired by the ramp-up schedules used for auxiliary objectives in consistency-based semi-supervised learning Laine & Aila (2016); Tarvainen & Valpola (2017), we adopt a smooth sigmoid scheduling function:

$$\lambda(e) = \frac{1}{1 + \exp[-k(e - e_0)]},$$

where e_0 controls the transition point, and k determines the growth rate.

3.5 Total Objective and Training Procedure

As shown in Figure 2(c), combining the time-step-wise predictive certainty with the training-stage-dependent separation weight, the final training objective is

$$L_{total} = (1 - \lambda(e))L_{\text{cls}} + \lambda(e) \sum_{t=1}^{T} D_t L_{\text{sep}}(s_t).$$

During training, the SNN first learns basic temporal classification under L_{cls}. As the predictive distributions become more informative, $\lambda(e)$ gradually increases, allowing the certainty-weighted separation loss to selectively enhance representations at reliable time steps. During inference, the auxiliary projection and feature separation components are discarded, and the original SNN readout is used to produce predictions at each time step without introducing additional computation for early decision-making.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

To examine the effectiveness and generalization, we evaluate our method on two benchmark datasets for keyword spotting.

Table 1: Comparison results on GSC V2.

Method	Acc@25 ↑	Acc@50 ↑	Acc@75 ↑	Acc@98 ↑
cumulative Bittar & Garner (2022)	8.20 ± 1.04	45.43 ± 1.94	83.56 ± 0.71	91.04 ± 0.42
spike rate Wu et al. (2018)	15.22 ± 1.29	56.89 ± 1.68	87.18 ± 0.15	91.95 ± 0.13
ED-sKWS Song et al. (2024a)	19.74 ± 0.26	59.61 ± 0.14	86.68 ± 0.19	91.56 ± 0.10
TET Deng et al. (2022)	22.12 ± 0.44	61.80 ± 0.61	87.38 ± 0.19	92.31 ± 0.12
STCL Qiu et al. (2024)	21.11 ± 0.46	55.10 ± 0.96	75.85 ± 1.01	77.02 ± 1.62
Ours	$\mathbf{22.71 \pm 0.11}$	$\mathbf{61.96 \pm 0.47}$	$\mathbf{87.62 \pm 0.02}$	$\mathbf{92.72 \pm 0.21}$

Google Speech Commands V2 (GSC V2) Warden (2018) consists of 105,829 approximately one-second speech utterances from thousands of speakers. In our experiments, we use the 35-class setting of Google Speech Commands V2, covering 35 keyword classes at a 16-kHz sampling rate. Its standardized splits and relatively balanced, short-duration recordings make it a widely used benchmark for evaluating efficient keyword spotting models.

Multilingual Spoken Words Corpus (MSWC) Microset Mazumder et al. (2021) contains 51 keyword classes in English and Spanish and is derived from the larger Multilingual Spoken Words Corpus, which is constructed from crowdsourced speech using automatic word-level alignment.

4.1.2 Implementation Details

Following ED-sKWS Song et al. (2024a), we extract log-Mel filter-bank (fbank) features from the audio signal using 40 filters, and each input utterance is divided into $T = 98$ time steps. All spiking neurons in the network are implemented using adaptive leaky integrate-and-fire (adLIF) neurons Bittar & Garner (2022). During training, we employ the standard surrogate-gradient method for SNNs, where the non-differentiable spiking function is approximated by a rectangular surrogate function with a range of $[-0.5, 0.5]$ for gradient computation Wu et al. (2018). We train all models for 200 epochs using the Adam optimizer with a learning rate of 0.01. All methods adopt the same network architecture and are trained under identical experimental settings.

4.1.3 Evaluation Metrics

We report the classification accuracy at different decision time steps t, denoted by $\mathrm{Acc@}t$, where $t \in \{25, 50, 75, 98\}$. At each time step t, the prediction is obtained by taking the $\arg\max$ of the cumulative classification output and comparing it with the ground-truth label. This evaluation directly measures how accurately the model can classify an utterance when only its prefix up to time step t is available.

For each experimental configuration, we independently train the model with three different random seeds and report the mean and standard deviation as $\mathrm{mean} \pm \mathrm{std}$.

4.2 Comparison with Existing Methods

We evaluate the proposed method on both GSC V2 and MSWC Microset, comparing it with cumulative supervision Bittar & Garner (2022), the spike rate loss of spatio-temporal backpropagation Wu et al. (2018), the cumulative temporal loss of ED-sKWS Song et al. (2024a), temporal efficient training (TET) Deng et al. (2022), and supervised temporal contrastive learning (STCL) Qiu et al. (2024). The results are reported in Tables 1 and Table 2.

Our method achieves the highest $\mathrm{Acc@}t$ at all evaluated time steps on both GSC V2 and MSWC Microset. This result indicates that the proposed feature separation objective improves early decision-making without sacrificing the accuracy of late decisions.

The cumulative and spike-rate objectives exhibit substantially lower accuracy at early time steps, which can be attributed to their respective formulations for modeling early logits or temporal firing statistics. In contrast, STCL Qiu et al. (2024) provides competitive early-time performance but exhibits a substantial degradation at later time steps. STCL treats representations of the same sample across different time steps, and representations of samples sharing a class label, as positive pairs, while the remaining pairs act as negatives, implicitly assuming that representations at different time

Table 2: Comparison results on MSWC Microset.

Method	Acc@25 ↑	Acc@50 ↑	Acc@75 ↑	Acc@98 ↑
cumulative Bittar & Garner (2022)	2.45 ± 1.72	35.40 ± 1.66	82.34 ± 1.78	82.05 ± 2.32
spike rate Wu et al. (2018)	8.39 ± 0.99	52.45 ± 1.73	90.77 ± 0.36	88.07 ± 0.47
ED-sKWS Song et al. (2024a)	10.15 ± 0.97	62.33 ± 4.09	81.91 ± 3.32	76.92 ± 4.39
TET Deng et al. (2022)	10.41 ± 0.30	73.73 ± 0.23	89.72 ± 0.05	89.72 ± 0.03
STCL Qiu et al. (2024)	10.38 ± 0.33	65.87 ± 1.90	78.22 ± 2.51	80.16 ± 2.32
Ours	$\mathbf{10.59 \pm 1.17}$	$\mathbf{73.85 \pm 0.82}$	$\mathbf{90.10 \pm 0.23}$	$\mathbf{90.19 \pm 0.24}$

Table 3: Ablation study of certainty-guided feature separation on GSC V2.

Method	Acc@25 ↑	Acc@50 ↑	Acc@75 ↑	Acc@98 ↑
Uniform Separation	20.77 ± 0.16	59.08 ± 0.45	85.17 ± 0.31	90.12 ± 0.23
Ours	$\mathbf{22.71 \pm 0.11}$	$\mathbf{61.96 \pm 0.47}$	$\mathbf{87.62 \pm 0.02}$	$\mathbf{92.72 \pm 0.21}$

steps are sufficiently comparable. This assumption is potentially problematic for speech, where the amount of discriminative information increases progressively over time. Enforcing strong attraction between early and late representations despite their substantially different information content may force semantically mismatched representations to become unnecessarily aligned.

4.3 Ablation Studies

We conduct ablation studies on GSC V2 to investigate the contributions of the certainty-guided weighting mechanism and the dynamic scheduling strategy.

4.3.1 Certainty-Weighted versus Uniform Separation

We first investigate whether using the predictive certainty D_t as a time-step-wise weight is necessary. Specifically, we apply the feature separation loss with equal strength at every time step, which does not distinguish between time steps according to the predictive reliability. The results are reported in Table **??**.

Uniform separation applies the same feature separation strength to all time steps, without accounting for the reliability of the corresponding predictions. This variant consistently underperforms our certainty-weighted formulation and even falls below the TET Deng et al. (2022) without feature separation at later time steps. This suggests that uniformly enforcing feature separation can negatively affect the final classification performance.

In contrast, incorporating D_t automatically suppresses the separation strength at low-certainty time steps while strengthening it at informative time steps. The consistent improvements across all decision time steps support the effectiveness of predictive certainty as a temporal weighting signal for feature separation.

4.3.2 Dynamic Weight Scheduling versus Fixed Weight

We next investigate how the training-stage-dependent separation coefficient affects performance. We compare several scheduling strategies on GSC V2, including fixed weights, monotonically increasing weights, sigmoid growth, rise-and-fall scheduling, and warmup-decay scheduling. In the fixed schedule, the coefficient is kept constant at its maximum value throughout training. In the decay schedule, it starts from this maximum and decreases linearly with the training epoch until it reaches zero at the end of training. The growth schedule does the opposite: it increases linearly from zero and reaches the maximum at the final epoch. The sigmoid-growth schedule changes smoothly from a value near zero to the maximum according to a sigmoid transition, centered at a specified transition epoch, with a steepness parameter determining how quickly the transition occurs. The rise-and-fall schedule maintains the maximum value during the first half of training and then decays it to zero. The warmup-decay schedule sets the coefficient to zero for an initial warmup period and then applies a linear decay.

Table 4: Different separation-weight schedules on GSC V2.

Schedule	**Acc@25** ↑	**Acc@50** ↑	**Acc@75** ↑	**Acc@98** ↑
sigmoid growth	22.71 ± 0.11	$\mathbf{61.96 \pm 0.47}$	$\mathbf{87.62 \pm 0.02}$	$\mathbf{92.72 \pm 0.21}$
fixed	22.27 ± 0.36	61.35 ± 0.22	86.50 ± 0.20	91.55 ± 0.15
rise fall	22.25 ± 0.32	61.13 ± 0.49	87.08 ± 0.21	92.02 ± 0.16
increasing	22.64 ± 0.53	61.16 ± 0.20	87.15 ± 0.21	92.29 ± 0.18
warmup decay	22.11 ± 0.43	60.22 ± 0.63	84.91 ± 0.38	91.28 ± 0.24

The corresponding results are reported in Table 4.

The sigmoid growth schedule achieves the highest performance at the three middle-to-late decision points. At the earliest time step, $\mathrm{Acc@25}$, the differences among the schedules are within their respective standard deviations, suggesting that their effects are relatively limited at this stage. This is expected because the amount of speech information available at very early time steps is inherently limited, leaving less room for the scheduling strategy to affect the single-step prediction.

The advantage of sigmoid growth becomes more evident at intermediate and late time steps. By gradually increasing the separation strength during training, it aligns the feature separation objective with the development of the SNN's predictive capability, allowing the accumulated output to reach high confidence earlier once sufficient information becomes available. In contrast, fixed and early-peaking schedules do not explicitly account for the evolution of the main classification task and consequently achieve lower performance at later time steps.

Overall, these results support the proposed coarse-to-fine training strategy: the feature separation objective should be suppressed while the basic classification mapping is being established, and its contribution should be gradually increased once the predictive distribution becomes sufficiently informative. This scheduling strategy allows the certainty-guided separation mechanism to operate when its temporal weighting signal is more reliable.

5 Conclusion

In this work, we presented a certainty-guided spiking feature separation framework for early decision-making in keyword spotting. The proposed method exploits the predictive distribution of the SNN at each time step to estimate the informativeness of intermediate representations and dynamically modulates feature separation accordingly. Furthermore, a dynamic weight scheduling strategy progressively increases the contribution of feature separation during training, allowing the auxiliary objective to better align with the evolving classification capability of the SNN. Experiments on GSC V2 and MSWC Microset datasets demonstrate that the proposed framework consistently improves recognition accuracy at early decision time steps while maintaining competitive final accuracy. Since the additional feature separation mechanism is only used during training and introduces no extra computation during inference, the proposed approach provides a lightweight way to improve the temporal discriminability of SNN representations. Overall, this work demonstrates the potential of certainty-guided representation learning for low-latency and energy-efficient speech recognition and provides a general framework that could be extended to other sequential tasks requiring reliable early decisions.

AI use statement

In this work, we used generative AI tools to assist with translation. We have not used generative AI tools for other tasks with required disclosure. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

Alexandre Bittar and Philip N Garner. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16:865897, 2022.

Paolo G Cachi, Sebastián Ventura, and Krzysztof J Cios. Improving spiking neural network performance with auxiliary learning. *Machine Learning and Knowledge Extraction*, 5(3):1010–1022, 2023.

Guangyao Chen, Peixi Peng, Guoqi Li, and Yonghong Tian. Training full spike neural networks via auxiliary accumulation pathway. *arXiv preprint arXiv:2301.11929*, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Seungwoo Choi, Seokjun Seo, Beomjun Shin, Hyeongmin Byun, Martin Kersner, Beomsu Kim, Dongyoung Kim, and Sungjoo Ha. Temporal convolution for real-time keyword spotting on mobile devices. *arXiv preprint arXiv:1904.03814*, 2019.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee micro*, 38(1):82–99, 2018.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Maxence Lasbordes, Daniele Falavigna, and Alessio Brutti. Splitformer: An improved early-exit architecture for automatic speech recognition on edge devices. In *2025 33rd European Signal Processing Conference (EUSIPCO)*, pp. 341–345. IEEE, 2025.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pp. 562–570. Pmlr, 2015.

Tzu-Quan Lin, Hung-yi Lee, and Hao Tang. Daisy: Data adaptive self-supervised early exit for speech representation models. *arXiv preprint arXiv:2406.05464*, 2024.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. Fastbert: a self-distilling bert with adaptive inference time. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 6035–6044, 2020.

Iván López-Espejo, Zheng-Hua Tan, John HL Hansen, and Jesper Jensen. Deep spoken keyword spotting: An overview. *IEEE Access*, 10:4169–4199, 2021.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Mark Mazumder, Sharad Chitlangia, Colby Banbury, Yiping Kang, Juan Manuel Ciro, Keith Achorn, Daniel Galvez, Mark Sabini, Peter Mattson, David Kanter, et al. Multilingual spoken words corpus. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

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.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Haonan Qiu, Zeyin Song, Yanqi Chen, Munan Ning, Wei Fang, Tao Sun, Zhengyu Ma, Li Yuan, and Yonghong Tian. Temporal contrastive learning for spiking neural networks. In *International Conference on Artificial Neural Networks*, pp. 422–436. Springer, 2024.

Aurélien Renault, Alexis Bondu, Antoine Cornuéjols, and Vincent Lemaire. Early classification of time series: A survey and benchmark. *arXiv preprint arXiv:2406.18332*, 2024.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, and Carlo Gatta. 441 yoshua bengio. fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 247, 2014.

Patrick Schäfer and Ulf Leser. Teaser: early and accurate time series classification. *Data mining and knowledge discovery*, 34(5):1336–1362, 2020.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Zeyang Song, Qianhui Liu, Qu Yang, Yizhou Peng, and Haizhou Li. Ed-skws: Early-decision spiking neural networks for rapid, and energy-efficient keyword spotting. *arXiv preprint arXiv:2406.12726*, 2024a.

Zeyang Song, Jibin Wu, Malu Zhang, Mike Zheng Shou, and Haizhou Li. Spiking-leaf: A learnable auditory front-end for spiking neural networks. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 226–230. IEEE, 2024b.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Shuai Wang, Dehao Zhang, Kexin Shi, Yuchen Wang, Wenjie Wei, Jibin Wu, and Malu Zhang. Global-local convolution with spiking neural networks for energy-efficient keyword spotting. *arXiv preprint arXiv:2406.13179*, 2024.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2246–2251, 2020.

Zhengzheng Xing, Jian Pei, and S Yu Philip. Early prediction on time series: A nearest neighbor approach. In *IJCAI*, number 11-17, pp. 2009, 2009.

Qu Yang, Qi Liu, and Haizhou Li. Deep residual spiking neural network for keyword spotting in low-resource settings. In *Interspeech*, pp. 3023–3027, 2022.

Emre Yılmaz, Özgür Bora Gevrek, Jibin Wu, Yuxiang Chen, Xuanbo Meng, and Haizhou Li. Deep convolutional spiking neural networks for keyword spotting. In *Proc. Interspeech 2020*, pp. 2557–2561, 2020.

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10): 905–913, 2021.

Ji Won Yoon, Beom Jun Woo, and Nam Soo Kim. Hubert-ee: Early exiting hubert for efficient speech recognition. *arXiv preprint arXiv:2204.06328*, 2022.

Neil Zeghidour, Olivier Teboul, Félix De Chaumont Quitry, and Marco Tagliasacchi. Leaf: A learnable frontend for audio classification. *arXiv preprint arXiv:2101.08596*, 2021.

Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128*, 2017.

Xiaochen Zhao, Chengting Yu, Kairong Yu, Lei Liu, and Aili Wang. Enhanced self-distillation framework for efficient spiking neural network training. *Advances in Neural Information Processing Systems*, 38:111546–111571, 2026.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020.