Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

*Digital Object Identifier 10.1109/XXXX.2022.1234567*

# Leveraging Beam Search Information for Confidence Estimation in E2E ASR

**Yichen Jia[1], Hugo Van hamme[1], Senior Member, IEEE**

[1]Department Electrical Engineering ESAT-PSI, KU Leuven, Belgium

Corresponding author: Yichen Jia (email: yichen.jia@kuleuven.be).

This research was supported by the Flemish Government under the FWO-SBO grant S004923N: NELF, and KU Leuven grant C24M/22/025.

**ABSTRACT** To estimate confidence for end-to-end Automatic Speech Recognition (ASR) systems, recent research has proposed Confidence Estimation Modules that incorporate features from the backbone ASR model. Most existing approaches, however, are architecture-dependent. In this paper, we propose the Score-Rank Confidence Estimation Module (SR-CEM), a lightweight module that leverages beam search information to generate token- and word-level confidence scores. Specifically, SR-CEM constructs features by combining the scores and ranks of tokens within a hypothesis. Experiments show that SR-CEM achieves effective calibration on both in-domain and out-of-domain English data. On the in-domain test set, it attains a Maximum Calibration Error of 4.50% and an Expected Calibration Error of 0.30% at the token level, significantly outperforming softmax confidence (20.04% and 1.75%, respectively). At the word level, SR-CEM achieves 8.17% and 0.35%, compared to 17.91% and 1.67% from softmax confidence. Furthermore, we demonstrate its robustness across hybrid and transducer ASR architectures with different decoding strategies, as well as on Dutch, noisy and conversational speech conditions. Our main finding is that SR-CEM is particularly effective in reducing Maximum Calibration Error, which is critical for reliable downstream use of ASR outputs, while maintaining architecture independence and generality across diverse evaluation conditions.

**INDEX TERMS** Automatic Speech Recognition, Beam Search, Confidence Estimation

## I. INTRODUCTION

CONTEMPORARY End-to-End (E2E) Automatic Speech Recognition (ASR) systems [1], [2] achieve strong recognition performance but yield biased confidence scores, typically overestimating certainty [3], [4]. Confidence derived from softmax probabilities [3] remains unreliable due to model overconfidence and biases from language models, beam search, and pruning [5]. Reliable confidence is crucial for downstream tasks such as active learning [6], semi-supervised learning [7], system combination [8], error correction [9], and user feedback.

Prior work on confidence estimation has progressed along two main directions. *Calibration methods* such as Temperature Scaling [4] and Matrix Scaling [10] apply post-hoc adjustments to model outputs using a small number of parameters. While computationally efficient, these methods assume a single calibration curve applies uniformly, limiting effectiveness when models exhibit both over- and underconfidence.

*Confidence Estimation Modules (CEMs)* learn to predict confidence from model features. Li et al. [5] proposed an MLP-based CEM for Listen, Attend and Spell using decoder hidden states and attention. Qiu et al. [11] extended this to word-level estimation with aggregated features including hidden states, attention distributions, and top-K scores. Wang et al. [12] developed a CEM for RNN-T incorporating acoustic and lexical features from encoder and prediction networks. Naowarat et al. [13] proposed letter-level features from CTC outputs. More recently, Aggarwal et al. [14] adapted confidence estimation for Whisper, and Ravi et al. [15] introduced lexical similarity-based features.

Despite their effectiveness, these CEMs share a critical limitation: *architecture dependence*. They rely on internal model representations (hidden states, attention weights, encoder outputs) that vary across architectures, requiring model-specific adaptations and hindering portability. Methods designed for attention-based models cannot be applied

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

to CTC-only or transducer systems without substantial modification.

ASR decoding inherently encodes rich confidence cues in token scores and ranking. Motivated by this, we introduce the *Score-Rank Confidence Estimation Module (SR-CEM)*, a lightweight model that relies solely on beam search outputs, avoiding dependence on architecture-specific representations. While prior work has used ranking [11] and top-K scores [11] as input features alongside architecture-specific representations, SR-CEM demonstrates that beam search information alone suffices for effective confidence estimation. This enables true architecture-agnostic deployment across various ASR architectures, such as hybrid CTC/Attention, attention-only, CTC-only, and RNN-T models.

Our contributions are threefold: 1) We identify and analyze the local-global mismatch in beam search that causes systematic confidence bias, particularly for non-rank-one tokens; 2) We propose SR-CEM, a lightweight architecture-agnostic CEM that relies solely on beam search information; 3) We provide comprehensive evaluation demonstrating consistent improvements across architectures, languages (English, Dutch), and conditions (clean, noisy, conversational), with particular effectiveness in reducing maximum calibration error.

## II. BACKGROUND

### A. Confidence in ASR

In large-vocabulary subword-based ASR systems [16], an input sequence $X = \langle x_1, \ldots, x_M \rangle$ is mapped to a subword hypothesis $Y = \langle y_1, \ldots, y_T \rangle$, which is concatenated into a word-level transcription $W = \langle w_1, \ldots, w_L \rangle$. We use "subword" and "token" interchangeably [16], with tokens drawn from a vocabulary $\mathcal{V}$ of size $V$.

At each decoding step $t$, the model produces a score vector $\mathbf{s}_t \in \mathbb{R}^V$. The softmax confidence for token $y_t$, $c_Y$ is

$$c_Y^{\text{softmax}}(y_t) = \frac{\exp(\mathbf{s}_t[y_t])}{\sum_{v=1}^{V} \exp(\mathbf{s}_t[v])}, \tag{1}$$

which we refer to as *softmax* confidence. It reflects the model's raw probability for the selected token, without calibration.

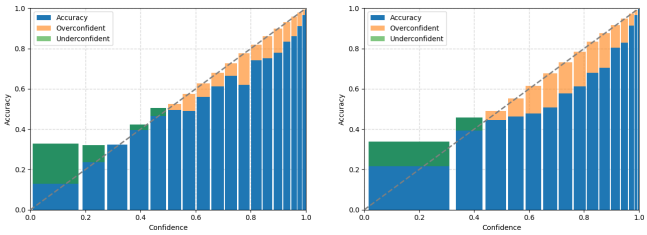
Word-level confidence scores are derived by aggregating the confidence values of the corresponding subword tokens. Specifically, for a word $w_l$ with tokens $\{y_{l,1}, \ldots, y_{l,Q_l}\}$, the word-level confidence, $c_W$ is computed as

$$c_W(w_l) = \text{agg}(c_Y(y_{l,1}), \ldots, c_Y(y_{l,Q_l})), \tag{2}$$

where agg denotes an aggregation function, such as the mean, minimum, maximum, or product. Following the approach in [11], we adopt the mean of token confidence as the confidence of a word. The mean normalizes for word length while representing the expected proportion of correct tokens.

### B. Confidence Calibration

Confidence calibration aims to produce scores $c \in [0, 1]$ that accurately reflect prediction correctness. Let $\mathcal{D} = \{y_1, \ldots, y_I\}$ denote a set of predictions (either tokens or words), where each prediction $y_i$ is associated with a confidence $c(y_i)$ and correctness $z(y_i) \in \{0, 1\}$, with 1 denotes correct and 0 incorrect.




(a) Token level (b) Word level

FIGURE 1: Reliability diagrams for softmax confidence scores on LibriSpeech test-clean.

The predictions are partitioned into bins based on their confidence scores using adaptive binning [17]. Unlike fixed binning schemes that pre-specify the number of bins $K$ and their boundaries, adaptive binning dynamically determines both the number of bins and their boundaries based on the data distribution. Let $B_k$ denote the $k$-th bin for $k = 1, \ldots, K$, where $K$ is data-dependent, and let $D_k = \{y_i \mid c(y_i) \in B_k\}$ represent the set of predictions assigned to this bin.

We define the joint distribution over confidence and correctness as $P_{\theta,\mathcal{D}}(c, z)$, where $\theta$ are the model parameters. Calibration error is quantified by Expected Calibration Error (ECE) and Maximum Calibration Error (MCE):

$$\widehat{\text{ECE}} = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{K} \left| \sum_{y_i \in D_k} c(y_i) - \sum_{y_i \in D_k} z(y_i) \right| \tag{3}$$

$$\widehat{\text{MCE}} = \max_k \frac{1}{|D_k|} \left| \sum_{y_i \in D_k} c(y_i) - \sum_{y_i \in D_k} z(y_i) \right| \tag{4}$$

Reliability diagrams [4] visualize calibration by comparing confidence and accuracy per bin. Fig. 1 shows token- and word-level diagrams for softmax confidence on LibriSpeech test-clean [18]. Overconfidence is a well-known issue in deep learning models [4], shown by the orange bars, whereas the green bars reflect underconfidence. With adaptive binning, both the number of bins and their characteristics (width and population) are determined by the algorithm. Wider bins may indicate regions where calibration is more uniform, while narrower bins capture regions requiring finer granularity for accurate calibration assessment.

### C. Beam Search and Scoring in ASR

Many modern ASR architectures use beam search [19], where hypotheses are expanded token by token and ranked by cumulative log probability scores. Beam search uses scores to select tokens, making them a natural representation of confidence. However, a critical mismatch exists: softmax confidence reflects local probability $p(y_t|y_{<t}, X)$,

while beam search selects based on cumulative score. This causes systematic underconfidence for tokens with low local probability that appear in the best hypothesis.

Let $r(y_t)$ denote the rank of $y_t$ in score vector $\mathbf{s}_t$ ($r = 1$ is highest). Softmax confidence satisfies:

$$c_Y^{\text{softmax}}(y_t) = \frac{\exp(\mathbf{s}_t[y_t])}{\sum_{v=1}^{V} \exp(\mathbf{s}_t[v])} \leq \frac{1}{r(y_t)}, \tag{5}$$

with equality only if top-$r$ scores are identical. In practice, $c_Y^{\text{softmax}}(y_t)$ is well below this bound, making correct non-rank-one tokens systematically underconfident (Fig. 1). This motivates our approach: SR-CEM uses rank to capture local competition, context scores for global quality, and top-K scores for uncertainty among alternatives. These features depend only on beam search dynamics, not internal model representations, enabling cross-architecture generalization while potentially being less effective with aggressive pruning or constrained decoding that limits alternative exploration.

## III. METHODOLOGY

Following the CEM framework, we construct features from beam search (token ranks and scores) and feed them into a lightweight module to predict confidence $\hat{c}$. We call this the *Score-Rank Confidence Estimation Module (SR-CEM)*.

### A. Token-Level SR-CEM

We design SR-CEM as a lightweight module consisting of one hidden layer with Rectified linear unit (ReLU) activation and a sigmoid output. The predicted confidence for token $y_t$ is

$$\hat{c}_Y(y_t) = \sigma(W_{Y,2} \cdot \text{ReLU}(W_{Y,1}\mathbf{f}_Y(y_t) + \mathbf{b}_{Y,1}) + \mathbf{b}_{Y,2}), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid, $W_{Y,1} \in \mathbb{R}^{64\times d_Y}$, $W_{Y,2} \in \mathbb{R}^{1\times 64}$, and $d_Y$ is the input feature dimension. The token-level feature vector $\mathbf{f}_Y(y_t)$ is described in the following text.

At each decoding step $t$, let $\mathbf{s}_t \in \mathbb{R}^V$ denote the score vector over the vocabulary $\mathcal{V}$ of size $V$. The selected token is $y_t \in \mathcal{V}$ with score $s_Y(y_t) = \mathbf{s}_t[y_t]$. The final hypothesis is $Y = \langle y_1, \ldots, y_T \rangle$, with total score

$$S(Y) = \sum_{t=1}^{T} s_Y(y_t). \tag{7}$$

For each token, the score $\mathbf{s}_t[y_t]$ provides basic confidence information. However, as discussed earlier, this score alone may be insufficient, since other tokens in the hypothesis also carry relevant information. To incorporate context, we define cumulative scores of preceding and succeeding tokens:

$$S_{<t} = \sum_{k=1}^{t-1} s_Y(y_k), \qquad S_{>t} = \sum_{k=t+1}^{T} s_Y(y_k), \tag{8}$$

so that $S(Y) = S_{<t} + s_Y(y_t) + S_{>t}$. Let $r(y_t)$ be the rank of $y_t$ within $\mathbf{s}_t$. Rank provides additional information beyond the raw softmax confidence. Following [11], we also include $\text{Topk}(t) \in \mathbb{R}^K$, the top-$K$ scores in $\mathbf{s}_t$ with $K = 4$, which capture uncertainty and competition among alternatives. We therefore define the token-level feature vector as

$$\mathbf{f}_Y(y_t) = \big[s_Y(y_t),\ r(y_t),\ S_{<t},\ S_{>t},\ \text{Topk}(t)\big], \tag{9}$$

where the 8 dimensions comprise: token score (1), token rank (1), preceding cumulative scores (1), succeeding cumulative scores (1), and top-k scores (4).

### B. Word-Level SR-CEM

We design SR-CEM to predict word-level confidence using a single hidden layer with ReLU activation followed by a sigmoid output:

$$\hat{c}_W(w_l) = \sigma(W_{W,2} \cdot \text{ReLU}(W_{W,1}\mathbf{f}_W(w_l) + \mathbf{b}_{W,1}) + \mathbf{b}_{W,2}), \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid, $W_{W,1} \in \mathbb{R}^{64\times d_W}$, $W_{W,2} \in \mathbb{R}^{1\times 64}$, and $d_W$ is the input feature dimension. The word-level feature vector $\mathbf{f}_W(w_l)$ is described in the following text.

To directly predict word-level confidence, we construct features from the corresponding token-level features. Let $w_l$ denote the $l$-th word in the hypothesis, consisting of $Q_l$ tokens: $w_l = \langle y_{l,1}, \ldots, y_{l,Q_l} \rangle$, where each token $y_{l,j} \in \mathcal{V}$ has an associated score vector $\mathbf{s}_{l,j} \in \mathbb{R}^V$.

The word score is defined as the sum of the token scores:

$$s_W(w_l) = \sum_{j=1}^{Q_l} \mathbf{s}_{l,j}[y_{l,j}]. \tag{11}$$

Analogous to the token-level case, we define preceding and succeeding scores:

$$S_{<l} = \sum_{k=1}^{l-1} s_W(w_k), \qquad S_{>l} = \sum_{k=l+1}^{L} s_W(w_k), \tag{12}$$

where $L$ is the total number of words in the hypothesis.

The word rank $r_l$ is taken as the maximum token rank within the word:

$$r_l(w_l) = \max_{1\leq j\leq Q_l} r(y_{l,j}), \tag{13}$$

where $r(y_{l,j})$ is the rank of token $y_{l,j}$ in $\mathbf{s}_{l,j}$. Among several aggregation strategies (mean, min, sum), the maximum rank performed best in preliminary experiments.

We define the word-level feature vector as

$$\mathbf{f}_W(w_l) = \big[s_W(w_l),\ r_l(w_l),\ S_{<l},\ S_{>l},\ Q_l\big], \tag{14}$$

where the 5 dimensions comprise: word score (1), word rank (1), preceding cumulative scores (1), succeeding cumulative scores (1), and token count (1).

### C. Data Preparation and Training

Informative features are collected during decoding for each token in the training data. The key input is the token score vector $\mathbf{s}_t$ at each decoding step of the winning hypothesis. After decoding, token and word correctness are labeled using Levenshtein alignment between ground-truth and predicted transcriptions.

Recognition errors include insertion, deletion, and substitution. Since deletions correspond to missing units, no confidence score can be assigned, and they are excluded from experiments. Remaining units are labeled as correct ($z = 1$) or incorrect ($z = 0$), with insertions and substitutions marked incorrect.

Training data are thus pairs of feature vectors and correctness labels. The objective is Binary Cross-Entropy (BCE) loss between the predicted confidence $\hat{c}$ and label $z$:

$$\mathcal{L}_{\text{BCE}}(\hat{c}, z) = -\left[z \cdot \log(\hat{c}) + (1 - z) \cdot \log(1 - \hat{c})\right]. \quad (15)$$

## IV. EXPERIMENTAL SETUP

Experiments are conducted with ESPnet [20], using custom code to extract decoding-time features without altering the original workflow. The proposed SR-CEM and baselines are implemented in PyTorch [21]. Code to reproduce the experiments will be made available.[1]

### A. Datasets and Backbone ASR

*LibriSpeech* [18] (LS) is a 1,000-hour English audiobook corpus with *clean* and *other* test splits. *Common Voice* [22] (CV) is a multilingual, crowd-sourced corpus with diverse accents. Due to transcription variability, ASR performance on CV is lower than LS [23], [24], making confidence estimation more challenging. We use only the U.S. accent subset from CV.

*Libri-Adapt* [25] (APT) is a benchmark dataset for domain adaptation in ASR. It extends LibriSpeech by introducing two dimensions of variability: microphone type and noise condition, while keeping the original transcriptions unchanged. The dataset includes six types of microphones and four types of background noise.

*Corpus Gesproken Nederlands* [26] (CGN) is a Dutch speech corpus covering spontaneous speech, interviews, and read speech from both the Netherlands and Flanders. We use the Flemish part with the train, test, and validation splits from [27] to evaluate cross-lingual robustness of confidence estimation.

*CHiME-6* [28] is a conversational speech corpus featuring spontaneous dinner party conversations with background noise. It provides a more realistic and challenging benchmark than read or prompted speech datasets.

In our experiments, we treat LS as the main dataset and use CV as a domain-mismatch dataset, as both the audio characteristics and transcripts differ substantially. When an ASR model is trained on LS and evaluated on CV, the performance degradation is even more pronounced. The APT corpus is used as the noisy dataset, where we fine-tune the ASR model and subsequently train and evaluate the CEMs. We use CHiME-6 as the conversational speech dataset, where we also fine-tune the ASR model on it, then we train and evaluate the CEMs.

The backbone ASR system is a *hybrid CTC/Attention* model [19] with 12 Conformer encoder and 6 Transformer decoder layers. Both use an attention dimension of 256, feed-forward dimension of 2048, and 4 attention heads. The CTC weight is 0.3. The model is trained from scratch on LS train-clean-360 for 40 epochs with Adam [29], using a learning rate of 4.0, achieving 4.6% WER and 5.6% TER on LS test-clean. For out-of-domain evaluation, we test on CV (U.S. English subset: 45.8% WER, 48.3% TER).

We employ an *RNN-Transducer (RNN-T)* model [30], in which the acoustic and prediction networks jointly generate scores for each encoder frame. It is a Conformer-Transducer model with 12 Conformer encoder blocks and a single-layer LSTM prediction network. The encoder uses an attention dimension of 256, feed-forward dimension of 2048, and 4 attention heads, with relative positional encoding, and convolution modules enabled. The joint network projects into a 320-dimensional space. A hybrid training objective is used with a CTC weight of 0.3. The RNN-T model is also trained from scratch on LS train-clean-360 for 40 epochs using Adam [29] (learning rate=4.0), with 6.1% WER and 7.5% TER on LS test-clean.

The vocabulary $\mathcal{V}$ (Sec. A) is built with Byte Pair Encoding on LS train-clean-360, with size 5,000.

### B. Baselines

Most existing methods are architecture-specific. Methods like [13] construct features with vocabulary-dependent dimensionality (impractical for our 5,000-unit vocabulary) and target CTC-only systems (our hybrid model uses CTC weight 0.3). Temperature Scaling [4] applies fixed adjustments unsuitable for systems with both over- and underconfidence.

We adopt: *Softmax*, *MLPCEM* and *Xformer* (adapted from [11]), *E2EXformer* [11], and *TruCLeS* [15] for CTC/RNN-T models.

For MLPCEM we follow [11], using three fully connected layers of sizes 128, 32, and 1. As configurations for Xformer and E2EXformer were not released, we explored several setups and selected compact variants: Xformer uses hidden dim 64, feed-forward dim 128, and 1 attention head; E2EXformer uses the same hidden dim with a single Transformer decoder layer. Model sizes are reported in Tab. 1.

All methods are trained with Adam using a learning rate of 0.001 and weight decay of $1 \times 10^{-4}$, batch size 128, and early stopping. Each dataset is split 8:2 into train/validation using a fixed seed. Training runs for up to 20 epochs, and the best validation model is selected.

All the aforementioned baselines rely on a decoder with hidden states and are therefore not applicable to CTC-only models. For CTC-only and RNN-T models, we adopt *TruCLeS* [15] as the baseline. TruCLeS is a recent CEM that leverages lexical similarity scores between hypotheses and reference candidates to enhance confidence estimation. Because the backbone ASR model has different hidden dimensions, we adopt the original setting for TruCLeS RNN-T. The LSTMs are configured with 256 hidden units per direction and an input dimension of 1024. The TruCLeS models are trained using the same framework described in the previous paragraph. The learning rate is set to 0.0001, following the original work. The number of parameters is 5.4M for TruCLeS-RNNT, 7.4M for TruCLeS-CTC.

[1] https://github.com/windskylionheart1023/Score_Rank_Confidence_Estimation_Module

TABLE 1: Number of parameters per model. For word-level MLPCEM and Xformer, sizes match token-level variants.

| | Token level | | | Word level | |
|---|---|---|---|---|---|
| | MLPCEM | Xformer | SR-CEM | E2EXformer | SR-CEM |
| Size | 70k | 83k | 0.6k | 100k | 0.4k |

### C. Efficiency and Deployment

SR-CEM's two-layer architecture with 0.6k (token) / 0.4k (word) parameters is 100-250× smaller than baselines. Model memory footprint is less than 5KB. Feature storage scales with utterance length (approximately 12KB for 150 tokens), which is modest compared to ASR memory requirements. Training converges in a few minutes (20 epochs on 20k utterances, single 3060 GPU). Inference adds less than 0.1ms latency per utterance since scores and ranks are already computed during beam search.

We recommend hidden dimension $h = 64$ (balances expressiveness and simplicity), learning rate 0.001, weight decay $10^{-4}$, and batch size 128. When training data contains predominantly well-calibrated high-confidence samples, consider using stratified or balanced sampling to ensure diverse batch compositions for more effective training.

### D. Beam search for RNN-T model

The seminal RNN-T work [30] describes inference as involving a summarization over different prefixes. In contrast, modern speech recognition toolkits, such as ESPnet [20], SpeechBrain [31], NeMo [32], and fairseq [33], perform beam search for RNN-T models without prefix search [34]. Concretely, for each encoder frame, when a new non-blank token is appended to a hypothesis, a score vector is produced that assigns probabilities to all tokens in the vocabulary. If the blank token is selected, no token is added to the hypothesis. Our proposed SR-CEM is designed for this beam search setting without prefix search, which is the widely adopted approach in modern speech processing toolkits.

### E. Evaluation Metrics

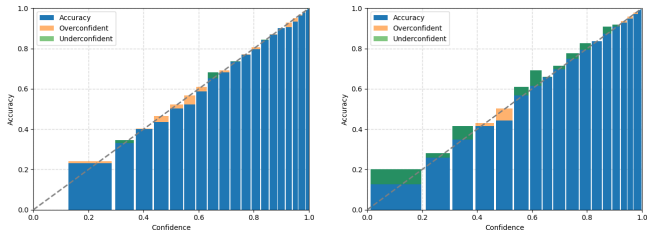
Following [5], [9], [11], [13], [14], we use several metrics. *Normalized Cross Entropy (NCE)* [35] evaluates how well scores align with correctness, ranging from $-\infty$ to 1 (higher is better).

To assess discriminative ability (separating correct from incorrect predictions), we report *AUC-ROC* and *AUC-PR* (denoted ROC and PR), both between 0 and 1, where higher is better. These metrics do not measure calibration quality but are included for comparison with prior work.

Calibration is measured by *Expected Calibration Error (ECE)* and *Maximum Calibration Error (MCE)* (Sec. B), computed using adaptive binning [17] to counter the skew of high-confidence scores. While ECE reflects average miscalibration, it is often dominated by already reliable predictions. We emphasize MCE, which captures worst-case errors in rare bins, which is crucial for downstream decisions where even small subsets of poorly calibrated scores can cause significant harm. Both ECE and MCE are reported in the tables as percentages for clarity.

TABLE 2: Token-level confidence results on LS test-clean and CV test sets. All CEMs trained on LS train-clean-100. ASR trained on LS train-clean-360

| Dataset | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| LS | Softmax | 0.301 | 0.919 | **0.996** | 1.75 | 20.04 |
| | MLPCEM | 0.320 | 0.891 | 0.993 | 0.43 | 21.22 |
| | Xformer | 0.298 | 0.875 | 0.993 | 0.37 | 12.06 |
| | *SR-CEM* | **0.383** | **0.923** | **0.996** | **0.30** | **4.50** |
| CV | Softmax | -0.029 | 0.853 | 0.926 | 15.99 | 34.77 |
| | MLPCEM | 0.126 | 0.840 | 0.911 | 12.91 | 25.62 |
| | Xformer | 0.125 | 0.836 | 0.913 | 13.48 | 29.65 |
| | *SR-CEM* | **0.248** | **0.866** | **0.931** | **9.54** | **19.59** |




(a) Token level (b) Word level

FIGURE 2: Reliability diagrams for SR-CEM confidence estimation on LibriSpeech test-clean at token and word levels (left and right respectively). SR-CEM trained on the LibriSpeech train-clean-100.

## V. RESULTS AND DISCUSSION

This section is organized as follows. We first present preliminary results with the hybrid CTC/Attention model on the in-domain and out-of-domain datasets. We then conduct an ablation study based on these experiments. Next, we extend the analysis to additional ASR architectures, including attention-only, CTC-only, and RNN-T models. We further evaluate our method on an out-of-domain dataset and on Dutch to explore language robustness. Finally, we assess the effectiveness of SR-CEM on noisy and conversational data.

### A. Token-Level Results

Tab. 2 shows SR-CEM outperforms all baselines on LS test-clean, achieving MCE of 4.50% compared to 20.04% for softmax (see Fig. 2a). Among baselines, MLPCEM improves NCE and ECE, while Xformer performs better on MCE. On out-of-domain CV, SR-CEM maintains the best performance (MCE 19.59%) despite poor ASR performance, while MLPCEM and Xformer achieve only 25.62% and 29.65%.

### B. Word-Level Results

Tab. 3 shows SR-CEM achieves the best word-level calibration on LS test-clean, reducing MCE from 17.91% (softmax)

TABLE 3: Word-level confidence estimation results on LS test-clean and CV test sets. All CEMs are trained on the LS train-clean-100. ASR trained on LS train-clean-360

| Dataset | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| LS | Softmax | 0.336 | **0.931** | **0.996** | 1.67 | 17.91 |
| | MLPCEM | 0.337 | 0.899 | 0.995 | 0.55 | 16.87 |
| | Xformer | 0.314 | 0.886 | 0.994 | 0.71 | 20.57 |
| | E2EXformer | 0.325 | 0.912 | **0.996** | 0.92 | 23.29 |
| | *SR-CEM* | **0.356** | 0.899 | 0.994 | **0.35** | **8.17** |
| CV | Softmax | -0.485 | **0.792** | **0.843** | 23.84 | 47.21 |
| | MLPCEM | -0.278 | 0.774 | 0.821 | 21.07 | 41.67 |
| | Xformer | -0.225 | 0.783 | 0.839 | 21.94 | 43.56 |
| | E2EXformer | -0.327 | 0.772 | 0.830 | 20.88 | 34.87 |
| | *SR-CEM* | **-0.198** | 0.778 | 0.804 | **18.62** | **32.78** |

TABLE 4: Leave-One-Feature-Out ablation study for token level on LS test-clean.

| Deleted feature | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|
| Score | 0.373 | 0.919 | 0.995 | 0.53 | 5.36 |
| Rank | 0.378 | 0.921 | **0.996** | 0.45 | 5.53 |
| Prec Score Sum | 0.377 | 0.920 | **0.996** | **0.30** | 6.36 |
| Succ Score Sum | 0.377 | 0.920 | **0.996** | 0.40 | 7.39 |
| TopK | 0.342 | 0.898 | 0.993 | 0.60 | 5.76 |
| *None* | **0.383** | **0.923** | **0.996** | **0.30** | **4.50** |

to 8.17% with lowest ECE (0.35%). While softmax achieves higher ROC/PR (0.931/0.996), SR-CEM provides better-calibrated scores (0.899/0.994) for threshold-based applications. On out-of-domain CV, SR-CEM reduces MCE from 47.33% to 32.78% despite all methods showing negative NCE. The word-level performance gap compared to token level (8.17% vs. 4.50% MCE) reflects the absence of top-K features, which limits feature expressiveness but maintains practical calibration quality.

### C. Ablation study

We conducted a Leave-One-Feature-Out ablation study on the LibriSpeech test-clean. As shown in Tab. 4 and Tab. 5, each feature contributes to performance gains, and removing any feature results in degradation. At the token level, removing Succeeding Score Sum has the largest impact on MCE, while removing rank shows the least effect. At the word level, removing the token count has the largest impact on MCE, while removing max rank shows the least effect.

TABLE 5: Leave-One-Feature-Out ablation study for word level on LS test-clean.

| Deleted feature | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|
| Score | 0.065 | 0.605 | 0.974 | **0.35** | 9.48 |
| Max Rank | 0.355 | 0.898 | **0.994** | 0.37 | 8.93 |
| Prec Score Sum | **0.356** | 0.898 | **0.994** | 0.38 | 10.44 |
| Succ Score Sum | 0.355 | 0.898 | **0.994** | 0.45 | 11.87 |
| #Token in word | 0.349 | 0.897 | **0.994** | 0.35 | 17.21 |
| *None* | **0.356** | **0.899** | **0.994** | **0.35** | **8.17** |

TABLE 6: Token- and word-level confidence results on the LS test-clean with attention-only decoding. All CEMs trained on the LS train-clean-100. ASR trained on LS train-clean-360.

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | 0.264 | 0.861 | 0.989 | 1.47 | 16.11 |
| | MLPCEM | 0.264 | 0.856 | 0.989 | 0.78 | 23.17 |
| | Xformer | 0.263 | 0.845 | 0.989 | **0.47** | 14.60 |
| | *SR-CEM* | **0.289** | **0.864** | **0.990** | 0.73 | **8.12** |
| Word | Softmax | **0.351** | **0.919** | **0.996** | 1.14 | 21.23 |
| | MLPCEM | 0.329 | 0.896 | 0.995 | **0.53** | 23.55 |
| | Xformer | 0.316 | 0.884 | 0.994 | 0.74 | 14.44 |
| | E2EXformer | 0.317 | 0.909 | **0.996** | 0.96 | 28.93 |
| | *SR-CEM* | 0.324 | 0.880 | 0.992 | 0.92 | **13.08** |

### D. Experiemtn with Attention-Only Decoding

The evaluation metrics depend on both predicted confidence scores and correctness labels. In hybrid ASR, correctness is determined by the final hypothesis combining attention and CTC scorers, while all baseline methods use only attention-based scoring. For fair comparison, we conduct an additional experiment with attention-only decoding, disabling the CTC scorer, extracting corresponding features, and retraining all models.

Tab. 6 shows SR-CEM outperforms baselines at token level (MCE 8.12% vs. 16.11% softmax) and maintains lowest word-level MCE (13.08% vs. 21.23% softmax), despite E2EXformer achieving competitive overall performance. While baselines benefit from the simplified setting, SR-CEM remains most effective at reducing maximum calibration error.

### E. Experiment with the CTC-Only Decoding

In the CTC-only setup, correctness is determined directly from the CTC best path without attention scoring. Since other baselines are designed for attention-based decoding, we compare SR-CEM only with softmax confidence and the TruCLeS baseline.

Tab. 7 shows CTC-only results comparing SR-CEM with softmax and TruCLeS. SR-CEM achieves best token-level performance (MCE 3.27% vs. 15.65% softmax, NCE 0.380 vs. 0.329) and word-level performance (MCE 7.36% vs. 18.95% softmax, 10.6% TruCLeS). While TruCLeS improves calibration over softmax, it shows weak discrimination, whereas SR-CEM balances both effectively.

### F. Experiment with RNN-T Model

We also test the methods in the RNN-Transducer (RNN-T) setting. The RNN-T model is trained from scratch on LS train-clean-360.

Tab. 8 shows RNN-T results. SR-CEM achieves best token-level performance (NCE 0.353, MCE 4.20% vs. 21.45% softmax) and lowest word-level MCE (11.95% vs. 25.58% E2EXformer, 25.56% softmax), confirming effectiveness across attention, CTC, and transducer architectures.

TABLE 7: Token and word-level confidence results on the LS test-clean with CTC-only decoding. All CEMs trained on the LS train-clean-100. ASR trained on LS train-clean-360.

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | 0.329 | 0.925 | **0.996** | 1.40 | 15.65 |
| | *SR-CEM* | **0.380** | **0.926** | **0.996** | **0.26** | **3.27** |
| Word | Softmax | 0.326 | 0.929 | **0.996** | 2.08 | 18.95 |
| | TruCLeS | 0.181 | 0.812 | 0.984 | 0.51 | 10.6 |
| | *SR-CEM* | **0.431** | **0.930** | **0.996** | **0.37** | **7.36** |

TABLE 8: Token and word-level confidence results for RNN-Transducer on the LS test-clean. All CEMs trained and LS train-clean-100. ASR trained on LS train-clean-360.

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | 0.105 | 0.894 | **0.992** | 3.14 | 21.45 |
| | MLPCEM | 0.300 | 0.874 | 0.989 | 0.64 | 20.69 |
| | Xformer | 0.289 | 0.863 | 0.988 | 0.56 | 11.56 |
| | TruCLeS | -0.023 | 0.635 | 0.965 | 2.62 | 10.76 |
| | *SR-CEM* | **0.353** | **0.903** | **0.992** | **0.39** | **4.20** |
| Word | Softmax | 0.179 | **0.908** | 0.994 | 2.57 | 25.56 |
| | MLPCEM | 0.311 | 0.873 | 0.991 | 0.77 | 20.58 |
| | Xformer | 0.301 | 0.865 | 0.990 | 0.88 | 14.64 |
| | E2EXformer | **0.337** | **0.908** | **0.995** | **0.57** | 25.58 |
| | TruCLeS | -0.026 | 0.625 | 0.964 | 2.56 | 13.37 |
| | *SR-CEM* | 0.142 | 0.768 | 0.984 | 0.71 | **11.95** |

### G. Experiment on Out-of-Domain Data

To assess target-domain training, we train SR-CEM on 20k CV utterances while keeping the ASR model unchanged. Tab. 9 shows SR-CEM achieves MCE of 6.67% (token) and 12.15% (word) compared to 34.77% and 47.21% for softmax, confirming benefits from domain-matched training data.

### H. Experiment on Dutch Language

Our method is language-agnostic, as it does not rely on language-specific designs. To validate this property, we evaluated it with a Dutch ASR model [27] trained on a large media subtitle corpus.

TABLE 9: Token and word-level confidence results on the CV test set. All CEMs trained on the CV validation set. ASR trained on CV train set.

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | -0.029 | 0.853 | 0.926 | 15.99 | 34.77 |
| | MLPCEM | 0.310 | 0.860 | 0.926 | 2.33 | 7.06 |
| | Xformer | 0.297 | 0.856 | 0.925 | 3.96 | 13.21 |
| | *SR-CEM* | **0.339** | **0.872** | **0.937** | **2.26** | **6.67** |
| Word | Softmax | -0.485 | 0.792 | 0.842 | 23.84 | 47.21 |
| | MLPCEM | 0.073 | 0.780 | 0.825 | 11.65 | 16.11 |
| | Xformer | 0.062 | 0.776 | 0.826 | 13.22 | 21.78 |
| | E2EXformer | 0.302 | 0.862 | 0.919 | 5.14 | 12.77 |
| | *SR-CEM* | **0.412** | **0.891** | **0.920** | **2.27** | **12.15** |

TABLE 10: Token and word-level confidence results for Dutch ASR on the CGN test set. All CEMs trained on CGN validation set. ASR trained on CGN train set [27].

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | -0.117 | 0.798 | 0.994 | 1.46 | 73.05 |
| | MLPCEM | 0.202 | 0.825 | 0.989 | 0.83 | 17.27 |
| | Xformer | **0.259** | 0.768 | 0.960 | 1.59 | 12.79 |
| | *SR-CEM* | 0.196 | **0.847** | **0.996** | **0.15** | **4.41** |
| Word | Softmax | -0.333 | **0.840** | **0.984** | 5.31 | 35.85 |
| | MLPCEM | 0.166 | 0.808 | 0.980 | 2.13 | 33.74 |
| | Xformer | 0.040 | 0.669 | 0.979 | **0.60** | 7.46 |
| | E2EXformer | 0.147 | 0.839 | 0.983 | 3.24 | 17.58 |
| | *SR-CEM* | **0.230** | 0.825 | 0.980 | 0.95 | **6.26** |

TABLE 11: Token and word-level confidence results on the APT test set. All CEMs trained on the APT validation set. ASR fine-tuned on APT train set.

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | -94.241 | 0.524 | 0.949 | 95.07 | 95.08 |
| | MLPCEM | 0.349 | 0.900 | **0.993** | 0.77 | 12.12 |
| | Xformer | 0.326 | 0.890 | 0.992 | 0.43 | 11.85 |
| | *SR-CEM* | **0.364** | **0.907** | **0.993** | **0.32** | **5.99** |
| Word | Softmax | 0.2484 | **0.926** | **0.996** | 2.54 | 24.76 |
| | MLPCEM | -0,416 | 0.535 | 0.930 | 6.78 | 82.66 |
| | Xformer | -0.395 | 0.533 | 0.931 | 7.00 | 81.33 |
| | E2EXformer | **0.394** | 0.924 | **0.996** | 0.65 | 9.50 |
| | *SR-CEM* | 0.346 | 0.891 | 0.992 | **0.64** | **5.58** |

At the token level, SR-CEM achieves the lowest calibration errors (ECE 0.15, MCE 4.41) while maintaining strong ROC and PR scores. At the word level, SR-CEM provides the best calibration with MCE of 6.26%, compared with 35.85% for softmax and 7.46% for Xformer. The results confirm that SR-CEM generalizes well across languages and consistently improves calibration quality.

### I. Experiment on Noisy Speech

To evaluate our method in practical noisy environments, we fine-tuned the backbone ASR on Libri-Adapt, achieving 6.3% WER and 7.4% TER.

The results in Tab. 11 show that confidence estimation on noisy data is more challenging than on clean speech. At the token level, SR-CEM achieves the best performance across all metrics (MCE 5.99%). At the word level, while E2EXformer shows strong discrimination, SR-CEM provides the most consistent calibration with the lowest ECE (0.64%) and MCE (5.58%). These findings highlight SR-CEM's robustness in practical noisy scenarios.

### J. Experiment on Conversational Speech

To further evaluate our method in a more real-word setting, we fine tune the backbone CTC/Attention ASR model on the CHiME-6 train set.

Tab. 12 shows SR-CEM's performance on conversational speech. SR-CEM achieves the best calibration at both token

TABLE 12: Token and word-level confidence results on the CHiME-6 test set. All CEMs trained on the CHiME-6 validation set. ASR fine-tuned on CHiME-6 train set.

| Level | Method | NCE↑ | ROC↑ | PR↑ | ECE↓ | MCE↓ |
|---|---|---|---|---|---|---|
| Token | Softmax | 0.015 | 0.784 | 0.838 | 13.63 | 24.69 |
| | MLPCEM | 0.173 | 0.793 | 0.846 | 3.49 | 22.65 |
| | Xformer | 0.183 | 0.789 | 0.841 | 3.38 | 12.36 |
| | *SR-CEM* | **0.207** | **0.798** | **0.850** | **1.93** | **7.51** |
| Word | Softmax | 0.189 | 0.828 | 0.892 | 10.79 | 23.59 |
| | MLPCEM | 0.216 | 0.817 | 0.871 | 3.69 | 19.87 |
| | Xformer | 0.219 | 0.810 | 0.867 | 3.88 | 10.00 |
| | E2EXformer | 0.218 | 0.814 | 0.869 | 3.98 | 14.12 |
| | *SR-CEM* | **0.252** | **0.829** | **0.885** | **3.42** | **7.46** |

level (MCE 7.51% vs. 24.69% for softmax) and word level (MCE 7.46%, ECE 3.42%), demonstrating robustness on realistic spontaneous speech alongside the read and noisy speech results.

### K. Discussion

Our evaluation reveals several consistent patterns. SR-CEM prioritizes calibration over maximal discrimination, achieving competitive ROC/PR while dramatically reducing MCE—a trade-off valuable for threshold-based applications. This pattern holds across all architectures (hybrid, attention-only, CTC-only, RNN-T), with CTC-only showing the best calibration (MCE 3.27%), likely because frame-independent assumptions reduce the local-global mismatch in autoregressive decoding.

Domain shift affects all methods, but SR-CEM maintains its relative advantage on out-of-domain CV (MCE 19.59% vs. 34.77% softmax), improving substantially when trained on the target domain (6.67%). Language-agnostic behavior is confirmed by Dutch results (MCE 4.41% token, 6.26% word) mirroring English performance. Acoustic robustness extends to noisy (5.99%/5.58%) and conversational (7.51%/7.46%) conditions, with conversational speech presenting the greatest challenge due to spontaneous disfluencies and background noise.

Architecture-dependent baselines show inconsistent performance. MLPCEM improves ECE but not MCE, while TruCLeS shows weak discrimination despite being applicable to multiple architectures. Our results highlight MCE's importance: while many methods achieve low average error, SR-CEM's consistent 50-70% MCE reduction addresses worst-case bins critical for practical deployment.

The word-level performance gap (8.17% vs. 4.50% token-level MCE) reflects the absence of top-K features. Future work could incorporate N-best list features or beam statistics, though this would come at a computational cost.

## VI. CONCLUSION AND FURTHER RESEARCH

We proposed SR-CEM, a lightweight confidence estimation module that leverages score and rank information from beam search decoding. Unlike previous methods, SR-CEM is independent of the ASR model architecture and relies solely on information available during decoding. Our experiments show that SR-CEM performs strongly on both in-domain and out-of-domain datasets, and consistently outperforms multiple baseline methods across diverse evaluation metrics. In particular, it is effective in reducing Maximum Calibration Error (MCE), which is critical for reliable downstream use of ASR outputs. To further validate its robustness, we evaluated SR-CEM against multiple baselines across different ASR architectures, datasets, languages, and noisy and conversational conditions. These results demonstrate both the effectiveness and the generality of the method.

SR-CEM has several limitations. It requires beam search decoding, as greedy decoding or aggressive pruning reduces feature informativeness. It fails when there is serious mismatch between training and test conditions, such as training on clean speech but testing on noisy conditions, or encountering out-of-domain data with new vocabulary and acoustic patterns (CV: MCE 19.59% vs. 4.50% in-domain). These mismatches cause beam search score distribution shifts that degrade calibration. SR-CEM relies on the single-best hypothesis, ignoring beam alternatives. When multiple hypotheses have similar scores, it cannot detect this uncertainty. Word-level calibration remains weaker than token-level (8.17% vs. 4.50% MCE) due to absent top-K features. While SR-CEM's architecture is language-agnostic, learned calibration curves are not. Cross-language deployment requires retraining.

Confidence estimation in ASR remains fragmented and lacks a unified framework. This work moves toward an architecture-agnostic approach, though extending confidence estimation across models is still an open direction. Future work could address these limitations through training on diverse speech data to learn more robust confidence patterns, incorporating N-best features or beam diversity measures, developing domain-adaptive calibration methods, and exploring context-aware word-level aggregation strategies. Although our experiments used a hybrid Attention/CTC model, the idea behind SR-CEM generalizes to setups that integrate multiple knowledge sources, such as lexical constraints or large language model scores.

## REFERENCES

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[3] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2017.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[5] Q. Li, D. Qiu, Y. Zhang, B. Li, Y. He, P. C. Woodland, L. Cao, and T. Strohman, "Confidence estimation for attention-based sequence-to-sequence models for speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6388–6392.

[6] T. Drugman, J. Pylkkonen, and R. Kneser, "Active and semi-supervised learning in asr: Benefits on the acoustic and language models," *arXiv preprint arXiv:1903.02852*, 2019.

[7] F. Kreyssig, "Active and semi-supervised learning for speech recognition," Ph.D. dissertation, 2024.

[8] G. Evermann and P. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. Speech Transcription Workshop*, vol. 27. Baltimore, 2000, pp. 78–81.

[9] M. Naderi, E. Hermann, A. Nanchen, S. Hovsepyan, and M. M. Doss, "Towards interfacing large language models with ASR systems using confidence measures and prompting," in *Proc. Interspeech 2024*, 2024, pp. 2980–2984.

[10] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration," *Advances in neural information processing systems*, vol. 32, 2019.

[11] D. Qiu, Q. Li, Y. He, Y. Zhang, B. Li, L. Cao, R. Prabhavalkar, D. Bhatia, W. Li, K. Hu *et al.*, "Learning word-level confidence for subword end-to-end asr," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6393–6397.

[12] M. Wang, H. Soltau, L. El Shafey, and I. Shafran, "Word-level confidence estimation for rnn transducers," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 1170–1177.

[13] B. Naowarat, T. Kongthaworn, and E. Chuangsuwanich, "Word-level confidence estimation for ctc models," in *Proc. Interspeech 2023*, 2023, pp. 3297–3301.

[14] V. Aggarwal, S. S. Nair, Y. Verma, and Y. Jogi, "Adopting whisper for confidence estimation," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[15] N. Ravi, V. Arora *et al.*, "Teles: Temporal lexeme similarity score to estimate confidence in end-to-end asr," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[16] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2023.

[17] Y. Ding, J. Liu, J. Xiong, and Y. Shi, "Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 4–5.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[19] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[20] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211.

[21] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[22] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[23] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[24] Y. Peng, J. Tian, B. Yan, D. Berrebbi, X. Chang, X. Li, J. Shi, S. Arora, W. Chen, R. Sharma *et al.*, "Reproducing whisper-style training using an open-source toolkit and publicly available data," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[25] A. Mathur, F. Kawsar, N. Berthouze, and N. D. Lane, "Libri-adapt: a new speech dataset for unsupervised domain adaptation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7439–7443.

[26] N. Oostdijk *et al.*, "The spoken dutch corpus. overview and first evaluation." in *LREC*. Athens, Greece, 2000, pp. 887–894.

[27] J. Poncelet *et al.*, "Leveraging broadcast media subtitle transcripts for automatic speech recognition and subtitling," *arXiv preprint arXiv:2502.03212*, 2025.

[28] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj *et al.*, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *CHiME 2020-6th International Workshop on Speech Processing in Everyday Environments*, 2020.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[30] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. ICML Workshop on Representation Learning*, 2012.

[31] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[32] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[33] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[34] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[35] M.-h. Siu, H. Gish, and F. Richardson, "Improved estimation, evaluation and applications of confidence measures for speech recognition." in *Eurospeech*, 1997, pp. 831–834.





**Yichen Jia** (Member, IEEE) received the B.S. degree in financial engineering from Sichuan University, Chengdu, China, in 2021, and the M.S. degree in statistics and data science from KU Leuven, Leuven, Belgium, in 2024.

He is currently a Ph.D. student at the (Processing Speech and Images) PSI Lab, Department of Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium. His current research interests include automatic speech recognition, confidence estimation, streaming ASR, and LLM-ASR.





**Hugo Van hamme** (Senior Member, IEEE) received the master's degree in engineering (burgerlijk ingenieur) from Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 1987, the M.Sc. degree from Imperial College London, London, U.K., in 1988, and the Ph.D. degree in electrical engineering from VUB, in 1992.

From 1993 to 2002, he was with LH Speech Products and ScanSoft, initially as a Senior Researcher and later as a Research Manager. Since 2002, he has been a Professor with the Department of Electrical Engineering ESAT-PSI, KU Leuven, Leuven, Belgium. His main research interests include automatic speech assessment, assistive speech technology, source separation, noise-robust speech recognition, models of language acquisition, and computer-assisted learning.