# SQEE: A Machine Perception Approach to Sensing Quality Evaluation at the Edge by Uncertainty Quantification

Shuheng Li[1] Jingbo Shang[1] Rajesh K. Gupta[1] Dezhi Hong[2]
[1]University of California, San Diego [2]Amazon
{shl060,jshang,gupta}@ucsd.edu,hondezhi@amazon.com

## ABSTRACT

Cyber-physical systems are starting to adopt neural network (NN) models for a variety of smart sensing applications. While several efforts seek better NN architectures for system performance improvement, few attempts have been made to study the deployment of these systems in the field. Proper deployment of these systems is critical to achieving ideal performance, but the current practice is largely empirical via trials and errors, lacking a measure of quality. Sensing quality should reflect the impact on the performance of NN models that drive machine perception tasks. However, traditional approaches either evaluate statistical difference that exists objectively, or model the quality subjectively via human perception.

In this work, we propose an efficient sensing quality measure requiring limited data samples using smart voice sensing system as an example. We adopt recent techniques in uncertainty evaluation for NN to estimate audio sensing quality. Intuitively, a deployment at better sensing location should lead to less uncertainty in NN predictions. We design SQEE, Sensing Quality Evaluation at the Edge for NN models, which constructs a model ensemble through Monte-Carlo dropout and estimates posterior *total uncertainty* via average conditional entropy. We collected data from three indoor environments, with a total of 148 transmitting-receiving (t-r) locations experimented and more than 7,000 examples tested. SQEE achieves the best performance in terms of the top-1 ranking accuracy—whether the measure finds the best spot for deployment, in comparison with other uncertainty strategies. We implemented SQEE on a ReSpeaker to study SQEE's real-world efficacy. Experimental result shows that SQEE can effectively evaluate the data collected from each t-r location pair within 30 seconds and achieve an average top-3 ranking accuracy of over 94%. We further discuss generalization of our framework to other sensing schemes.

## CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; • **Computing methodologies** → **Neural networks**.

## KEYWORDS

Speech Sensing, Sensing Quality Evaluation, Uncertainty Quantification


*SenSys '22, November 6–9, 2022, Boston, MA, USA*

ACM ISBN 978-1-4503-9886-2/22/11.
https://doi.org/10.1145/3560905.3568534

**ACM Reference Format:**
Shuheng Li, Jingbo Shang, Rajesh K. Gupta, Dezhi Hong. 2022. SQEE: A Machine Perception Approach to Sensing Quality Evaluation at the Edge by Uncertainty Quantification. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3560905.3568534

## 1 INTRODUCTION

Neural network (NN) models have demonstrated potential in smart sensing systems such as voice sensing [26, 39], RF-based sensing [25, 52] and wearable device sensing [16, 19]. Most of existing works focus on the perception procedure, that is, to improve the effectiveness of these systems by designing better NN architectures. For example, various NN models have been proposed to cope with sampling frequency difference [13], data noise [27], sensing device variation [43] and sensory fault tolerance [51]. However, very few attempts have been made to investigate how to effectively deploy such sensing systems in the field to achieve ideal sensing quality.

As performance of NN models hinges on quality of sensory signal [53], it is crucial to deploy a sensing device at the best location. The most straightforward and accurate way to find the best sensing location with regard to machine perception – the machine learning problem(s) associated with the sensing modality – is by carrying out an exhaustive search among possible locations. To make fair comparison, the same set of signal needs to be transmitted from the same spots selected within the sensing area. The best sensing location is naturally defined as the one with the best average performance on all the data received at this location. However, this requires large number of examples to be tested for every possible combination of transmitting-receiving (t-r) locations, which can take days of experimentation. This is impractical at scale.

In this paper, we seek an efficient sensing quality measure for machine perception on edge devices with only a limited amount of data. Without loss of generality, we consider smart voice sensing system as an example, as commercial products like Google Assistant, Amazon Alexa and Tesla Voice Command are widely used in daily lives. We focus on the machine perception task of speech recognition, which involves sensing via a microphone array and perception via pretrained NN models that can be deployed either on the cloud or at the edge.

To be effective, the sensing quality should reflect the impact on the performance of NN models. There exists a variety of measures of audio sensing quality, but none of them are designed for NN-driven machine perception. Traditional methods such as structural similarity measure [17] quantify audio quality statistically, while standardized algorithms like PESQ [36] evaluate speech quality from the view of human auditory system. NN models use a large

number of parameters that are trained to naturally function as filters, enhancer or feature extractors. The underlying mechanism is fundamentally different from previous quality evaluations and provides us an opportunity to improve quality of sensing.

We adopt uncertainty evaluation approaches to estimate sensing quality. We define *uncertainty* as a quantifiable attribute that describes the doubt about the validity of a measurement [7]. Given a trained machine learning model, the only source of uncertainty in the system is sensing data quality imperfection, which suggests that sensing quality can be represented by model uncertainty on sampled data. In the machine learning community, early works model uncertainty as a belief score using Bayesian framework [22]. To avoid the complexity of Bayesian NN structure, model ensemble approaches such as Monte Carlo dropout (MC-dropout) [10] and deep ensembles [23] are used to approximate Bayesian frameworks. We study the potential of quantifying sensing quality through model uncertainty evaluation by analyzing a wide range of variants with different MC-dropout implementation and uncertainty estimations.

We design SQEE, Sensing Quality Evaluation at the Edge for NN models, that uses last layer MC-dropout for model ensemble and approximates *total uncertainty* via average token-level conditional entropy. We propose a criterion for choosing key hyper-parameters of SQEE, namely, dropout rate and model ensemble size for cloud and edge deployment. We further implement SQEE in our testbed environments to study its real-world efficacy by comparing the trade-off between quality evaluation accuracy and time efficiency. Experiment results suggest that SQEE can achieve decent top-2/top-3 ranking accuracy better than $94\%$ with only $2$ examples tested for each t-r location pair, which makes SQEE capable of efficiently evaluating data from each t-r pair within $30$ seconds.

The contributions of this work are as follows:

(1) We establish a framework for evaluating sensing quality as perceived by NN models that drive machine perception tasks. We use voice sensing as an example and conducted experiments in $3$ different indoor environments, including $148$ different deployments in total and more than $7000$ examples as our testbed. The evaluation focuses on top one ranking accuracy when limited data is available.
(2) We study the effectiveness of traditional signal quality evaluation approaches, including statistics-based metrics and subjective measures. We further evaluated the potential of utilizing uncertainty measurements for sensing quality evaluation.
(3) We present SQEE, a Sensing Quality Evaluation for NN models that approximates *total uncertainty* using MC-dropout model ensemble. We deployed SQEE that supports both online and offline evaluation on edge devices to study its efficacy with regard to accuracy and time efficiency.

# 2 SENSING QUALITY EVALUATION IS MORE DIFFICULT THAN YOU THINK

We first establish a benchmark to evaluate sensing quality measures in the new context of machine perception and experiment on existing voice quality evaluation algorithms. Our experiment results demonstrate that neither of the approaches can correctly reflect the sensing quality for machine perception.

## 2.1 Benchmark Dataset Collection

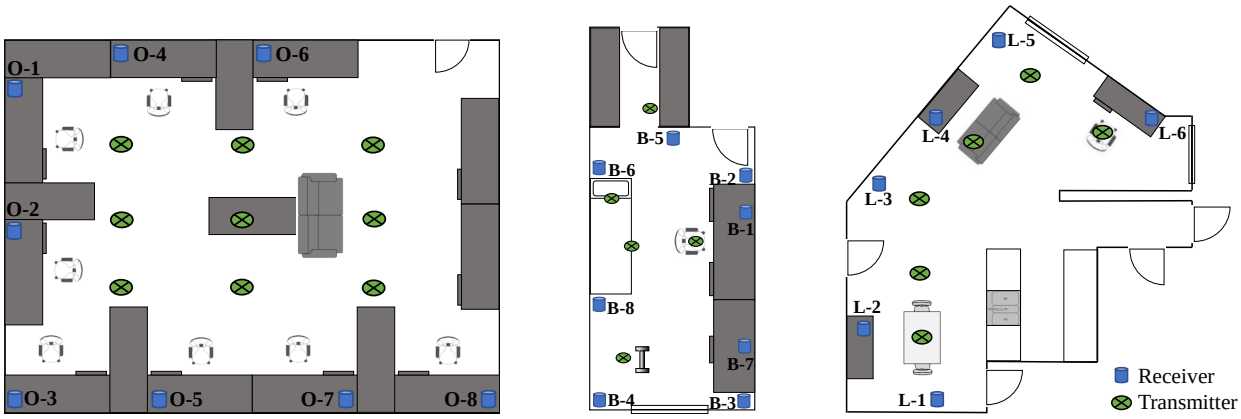
We deploy speech sensing systems in different indoor environments to collect benchmark datasets. In order to emulate real-world usage, at each candidate sensing location, we collect speech signal transmitted from a few representative locations within the sensing area using microphones. We test three different indoor environments: *office*, *bedroom* and *living room*. The deployments are illustrated in Figure 1 (a)-(c), respectively.

We mark the locations of receiver microphones using blue cylinders. We use a 4-mic ReSpeaker[1] as the sensing device, which consists of a Raspberry Pi 4 model B and a 4-microphone array located on top of it. It emulates household smart speakers like Alexa and Google Voice Assistant for both of them deploy microphone on the top of the devices. Considering that they are all powered by AC sockets, the candidate deployment locations for ReSpeaker are the same as other household smart speakers. We also use the same ReSpeaker device to implement SQEE at the edge for experiments on real-world efficacy in Section 4.5. As marked in Figure 1, we place microphones at locations where AC power sockets are available.

Green crossed circles represent the transmitting locations chosen based on common usage in the environment. To avoid variation caused by human speaking, we play through a bluetooth speaker the same set of recorded clean speech audio sampled from a speech recognition dataset. We use a Mifa A1 speaker[2] as the transmitter to emulate directional human speech. It is noteworthy that human speech is transmitted at the height of mouth, so we set the height of the speaker to mouth height when standing, sitting or lying down correspondingly.

For the *office* environment, we assume the whole sensing area is being actively used by occupants and voice commands to smart building controller are uniformly transmitted from the sensing area. To emulate this, we consider a $3\times3$ grid for transmitter deployment locations to cover the area, in which each location is $6$ feet away from the adjacent locations from both directions. The orientation of the speakers are fixed towards the center of the room. There are $8$ tables where sockets are available, which we take as the candidate locations for the microphone. For the *bedroom* environment, we consider a single person living in the bedroom, which makes transmitted voice command nonuniformly distributed in the sensing area but concentrated at representative spots instead. We deploy the transmitter accordingly to $5$ possible device usage situations including working, exercising, sitting, lying and crafting. The orientation that we deploy the transmitter also emulates user facial direction in these situations. Together with $8$ possible locations where AC power is available for sensor deployment, we collect data from $40$ t-r pairs in this environment. For the *living room* environment, unlike the previous two environments, the shape of the living room is irregular and its usage condition is also more complicated. The whole sensing area can be divided into zones for dining, cooking, TV watching and reading, among which we pick $6$ locations for the sensor and $6$ locations for the transmitter, resulting in $36$ pairs of t-r locations. We do not consider deployments around the kitchen due to power is limited for the sensor and the noise when cooking is usually much louder, which is different from other usage

[1] https://respeaker.io/4_mic_array/
[2] https://www.mifa.net/en/speakers/A-series/A1-Outdoor-Wireless-Speaker




(a) Office (b) Bedroom (c) Living room

**Figure 1: Deployment plan of (a) *office*, (b) *bedroom* and (c) *living room*. Blue cylinders denote the candidate locations of the receiver and green crossed circles denote the locations of the transmitter. The sensing quality of each blue location is determined using all the data transmitted from each green transmitting location.**

situations in livingroom.

For simplicity, we note the locations in *office*, *bedroom* and *living room* with O-1 to O-8, B-1 to B-8 and L-1 to L-6 accordingly.

We randomly select 50 examples from the testing set of LibriSpeech dataset [32]. As a collection of approximately 1,000 hours of audiobooks, the 960-hour training set of LibriSpeech is commonly used for pretraining large scale NN models. The transmitted data is henceforth sampled from the same distribution as the training data of the NN model, which leaves environmental variances during signal propagation to be the only source that degrades the signal quality. To match the data property for NN model pretraining, our microphone records at 16kHz, and we merge audio collected by the 4-microphone array to create monaural audio data.

## 2.2 Ground-Truth Sensing Quality Verification

We verify the ground truth sensing quality ranking to establish the comparison pipeline. In each environment, we denote all the collected audio data as $\boldsymbol{X}$ and denote the set of data collected with deployment on the $i$-th transmitting and the $j-$th receiving location as $\boldsymbol{X}_{(ij)}$. A speech recognition model $\boldsymbol{f}: \boldsymbol{x} \rightarrow \boldsymbol{y}$ maps the input speech audio $\boldsymbol{x}$ to the target natural language sequence $\boldsymbol{y}$. It is a similar perception subject as human auditory systems. Intuitively, an edge device deployed at the best sensing location collects speech data with the best quality in term of speech recognition performance. The most accurate way to measure sensing quality requires trying all the data collected at this location.

As a result, we use word error rate (WER), a metric for quantifying the performance of a speech recognition system as the ground-truth evaluation of the impact of sensing data on NN models. WER works by first aligning the recognized sequence and the reference sequence, and then computing $S, D, I, C$, the number of substitutions, deletions, insertions and correct words. WER is then calculated as $WER = \frac{S+D+I}{S+D+C}$. For each $j-$th sensing location, the quality score is obtained from the average WER with regard to $\boldsymbol{X}_{(:j)}$, namely, all the data collected here. We can further obtain the ranking of these locations using their quality scores, which we take as the ground-truth ranking.

## 2.3 Benchmark Evaluation

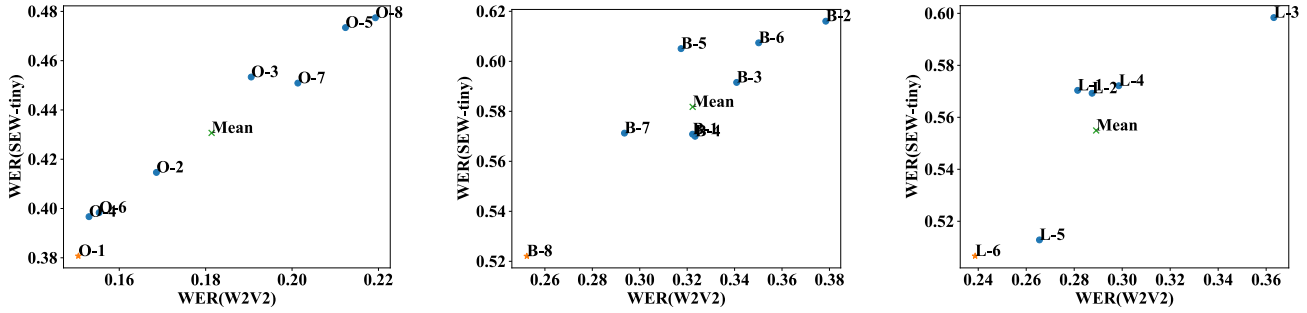
A smart sensing system can be implemented with NN models deployed either on the cloud or at the edge. A desired sensing quality evaluation algorithm should be efficient in both cases. It is common to use powerful cloud server for inference using bigger models when network connection is good. On the other hand, stable network connection is not guaranteed at all times and it is crucial to have a local pretrained NN model at the edge. Due to the limited memory and compute resource of the edge devices, the architecture and the size of the edge model needs to be carefully decided.

Specifically, we use Wav2Vec2 (W2V2) [3] as the cloud speech recognition model. W2V2 is a SOTA speech recognition model that learns representation of speech data using self-supervised learning. To deploy W2V2 at the edge, researchers have proposed a tiny version of W2V2 with only one eighth of the parameters and 4 times speed-up, but the performance is approximately 10 times worse than the base model in term of WER.

In our work, we use the tiny version of Squeezed and Efficient Wav2Vec (SEW-tiny) [49] model at the edge instead. Compared to W2V2-tiny, SEW-tiny utilizes numerous strategies to improve the performance while maintaining similar running time as W2V2-tiny.

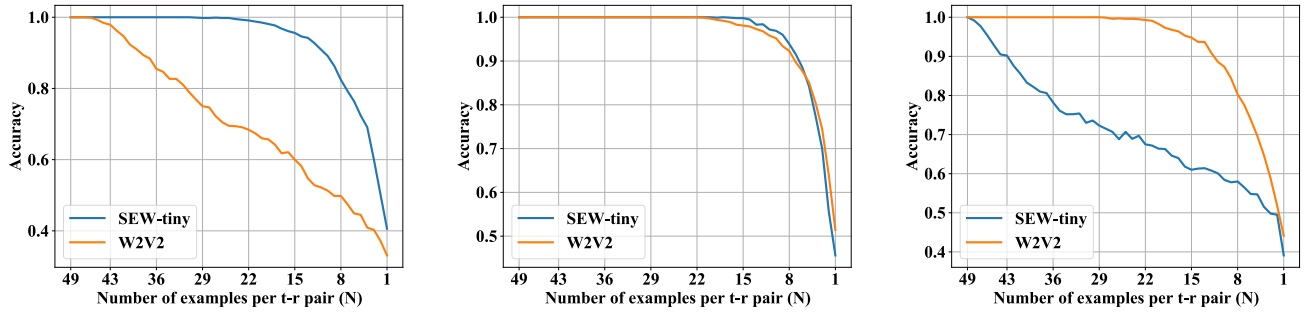
Both W2V2 and SEW-tiny are pretrained using LibriSpeech training set and fine-tuned for speech recognition tasks. W2V2 is pretrained using 960-hour data, while SEW-tiny uses 100-hour data. W2V2 is the most representative model in speech recognition. It is the best single model on the LibriSpeech benchmark dataset and further achieves the overall best performance using fine-tune strategies [54]. Since W2V2 and SEW-tiny can be optionally combined with language models, their uncertainty representations can also be generalized to other NN-based speech recognition models. We introduce the details in Section 3.3.

The ground-truth sensing quality given by WER is shown in Figure 2, where each point in circle represents the WER performance of W2V2 and SEW-tiny on all the data collected at each sensing location, among which the best location is highlighted as star in orange. To point out the expected performance when the sensing location is randomly chosen, the average WER of all the sensing




(a) Office (b) Bedroom (c) Living room

**Figure 2: Ground-truth Sensing Quality. The coordinate of each point represents the average word error rate (lower is better) of W2V2 and SEW-tiny for data received at each location. We note the best sensing location with orange star points and the expected performance if the location is randomly chosen with green crosses.**




(a) Office (b) Bedroom (c) Living room

**Figure 3: WER Measurement Accuracy. We visualize the top-1 accuracy when 1000 random sets of $N$ examples are tested.**

locations is noted by green cross. The best sensing location for *office*, *bedroom*, *living room* is O-1, B-8 and L-6 respectively and the result is consistent for both of the two NN models. The ground truth is defined to be model-specified. The reason that W2V2 and SEW-tiny have the same best sensing location might be they share similar backbone NN structure and they are trained on the same dataset. If one manages to find the best sensing location in the environment, it is expected that average word error rate can be reduced from 26.6% to 21.5%, compared with a randomly chosen location, which justifies that it is of great importance to evaluate sensing quality prior to system deployment. In addition, although we can observe explicit correlation between the performance of the two NN models, the rankings of all the locations at no time are the same. It suggests we evaluate signal quality subjectively from the view of specified NN models.

Obtaining the ground-truth ranking using WER requires a sufficient amount of data tested for variance reduction, which can be highly time-consuming, demanding repeated human labor exhausting all possible t-r location pairs. Since a cumbersome audio data collection process is also prone to random interruptions in the environment such as special vehicles passing by or sensor failure, recollection may also be inevitable. To this consideration, we focus on evaluating different speech quality measurements with only limited amount of speech audio collected for each t-r location pair. Let $g : (\boldsymbol{f}, \boldsymbol{x}) \rightarrow s$ denote a speech quality measurement that gives a score $s$ based on model $\boldsymbol{f}$ and input signal set $\boldsymbol{x}$. Suppose we can only afford to collect $N$ examples from each t-r pair, our evaluation pipeline emulates such process of collecting $N$ examples and repeats for a total of $k$ times. Since we are only interested in the location with the best speech quality in real-world scenario, we compare the best location given by $g$ and the best location according to ground-truth WER in the previous subsection. We use $\frac{\hat{k}}{k}$ as the accuracy of $g$ with $N$ examples available, where $\hat{k}$ denotes the total number of correct best locations among $k$ times.

We illustrate how WER performs with respect to the number of examples $N$ in Figure 3. Notice that the x-axes of these figures are in descending order for we mainly focus on how top-1 accuracy degrades with the number of tested examples decreasing. The experiment results suggest the existence of great room for further improvement when the amount of testing examples is limited. The general trend is that WER begins to rank the best location incorrectly when $N < 22$, when $N = 10$ the accuracy is around 90% but it drops drastically to around 50% for *bedroom* and to 40% for *office* and *living room* when only one example is tested for each t-r pair. Two exceptions occur for W2V2 on *office* and SEW-tiny on *living room* where the performance of WER degrades much earlier. However the performance degradation of the two curves is more gentle and the final $N = 1$ performance is comparable to the rest of four cases. It is because in *office*, the ground truth sensing quality

**Table 1: Result of Existing Methods. We use the entire set of data collected at each sensing location for evaluation. The best locations given by the ground-truth best sensing location and existing methods are bolded.**

| *office* | SNR | SSIM | PESQ | DNSMOS | STOI |
|---|---|---|---|---|---|
| **O-1** | 17.57 | 0.446 | 1.882 | 1.370 | 0.277 |
| O-2 | 17.49 | 0.458 | 1.796 | 1.285 | 0.278 |
| O-3 | 17.99 | 0.458 | 1.805 | 1.340 | 0.314 |
| O-4 | **19.29** | 0.438 | **1.907** | **1.479** | 0.310 |
| O-5 | 18.26 | 0.454 | 1.820 | 1.423 | 0.247 |
| O-6 | 17.51 | 0.447 | 1.847 | 1.320 | 0.273 |
| O-7 | 17.44 | 0.460 | 1.804 | 1.302 | 0.306 |
| O-8 | 16.46 | **0.467** | 1.803 | 1.293 | **0.366** |

| *bedroom* | SNR | SSIM | PESQ | DNSMOS | STOI |
|---|---|---|---|---|---|
| B-1 | 14.17 | 0.507 | 1.788 | 1.277 | 0.326 |
| B-2 | 12.71 | 0.511 | 1.698 | 1.293 | 0.207 |
| B-3 | 15.12 | 0.509 | 1.735 | 1.290 | **0.399** |
| B-4 | 13.31 | **0.512** | 1.713 | 1.260 | 0.281 |
| B-5 | **15.96** | 0.510 | 1.741 | 1.293 | 0.343 |
| B-6 | 12.03 | 0.511 | 1.724 | 1.351 | 0.269 |
| B-7 | 14.47 | 0.511 | 1.788 | 1.318 | 0.349 |
| **B-8** | 14.20 | 0.509 | **1.797** | **1.391** | 0.323 |

| *living room* | SNR | SSIM | PESQ | DNSMOS | STOI |
|---|---|---|---|---|---|
| L-1 | 20.30 | **0.473** | 1.804 | 1.492 | 0.323 |
| L-2 | 23.09 | 0.449 | 1.827 | 1.571 | 0.293 |
| L-3 | 22.47 | 0.436 | 1.844 | **1.680** | 0.258 |
| L-4 | 22.19 | 0.448 | 1.787 | 1.526 | 0.316 |
| L-5 | 24.17 | 0.459 | 1.914 | 1.618 | **0.407** |
| **L-6** | **26.01** | 0.455 | **1.930** | **1.680** | 0.396 |

of location O-1, O-4 and O-6 have similar WER score for the cloud model; and in *living room*, L-5 and L-6 have similar WER score for the edge model as shown in Figure 2. It henceforth makes these locations especially difficult to be distinguished by WER.

## 2.4 Review of Existing Methods

We review several existing speech quality measurements including statistics-based signal quality metrics that evaluate signal quality objectively using statistics quantities and perceptual speech quality evaluations that predict human judgement score. The rest of related work will be discussed in Section 6.

*2.4.1 Signal-to-Noise Ratio.* SNR is the most straightforward criterion that compares the ratio of signal power to noise power. In our experiment, we use Waveform Amplitude Distribution Analysis (WADA)-SNR, which assumes that the amplitude distribution of clean speech can be approximated by the Gamma distribution, and the additive noise signal is Gaussian [21].

*2.4.2 Structural Similarity Measure.* Structural Similarity Measure (SSIM) [17] is an entirely statistics-based measure. It evaluates three statistically measured differences between the degraded signal and the original signal: *luminosity*, *contrast* and *structure* that respectively measure the similarity between mean value, standard deviation and correlation. SSIM is derived as the exponential multiplication of the three measured statistical similarities.

*2.4.3 Short-time objective intelligibility (STOI).* STOI is an objective algorithm that measures the average intelligibility, i.e. the percentage of correctly recognized words by users [45]. STOI requires both the original signal and the reference signal and calculates the correlation coefficient between the processed temporal envelopes. The input signals are firstly TF-decomposed by an octave filterbank, segmented into short-time windows, normalized to match listening level difference, clipped and then compared by means of a correlation coefficient.

*2.4.4 Perceptual Evaluation of Speech Quality (PESQ).* PESQ [36] is a standardized algorithm that measures speech quality as perceived by humans and it models mean opinion score (MOS) with a scoring scale from -0.5 (bad) to 4.5 (excellent) using full-reference of the original signal. The original and the degraded signal are individually equalized to the same listening level and filtered to emulate the response of the receiving device. The signals are then aligned in the time axis and auditory transformed to a spectra in frequency-loudness domain in order to capture distortion perceived by human listeners. PESQ evaluates the loudness spectra difference between two signals and accumulates such audible errors over the time and frequency axes with $L_p$ norm. It distinguishes the symmetrical disturbance and asymmetrical disturbance and combines them linearly with learnt weights.

*2.4.5 DNSMOS.* DNSMOS is a non-intrusive perceptual objective speech quality measurement [38]. DNSMOS utilizes NN model trained by multi-stage self-teaching using human judgement MOS scores. It is originally designed for evaluating and ranking noise suppressors.

## 2.5 Existing Methods Are Ineffective

We evaluate the accuracy of above mentioned approaches in our benchmark dataset. For reproducibility, we use public Python implementation of these methods in our experiment. The estimated sensing quality estimated using the entire set of data collected at each sensing location is summarized in Table 1. We highlight the best locations given by these measurements and also the ground-truth best locations of the three environments in bold.

SNR ranks the best location correctly only in *living room* while SSIM and STOI fail to detect the best location in any of the environments. The three algorithms quantify noise level in data objectively by estimating statistics difference between the original clean signal and the received signal. However, NN models can usually be resistant to noise owing to the vast amount of parameters, but the effectiveness is inconsistent and the underlying mechanism is too complicated to be quantified. The same noise ratio or structural dissimilarity of different input data may not impact NN model performance equally, leading to the poor performance of the two existing approaches. Although PESQ and DNSMOS are the best approach among the five and they correctly find the best location in *bedroom* and *living room*, they cannot fully capture the sensing quality as perceived by NN models and fail to find the best location in *office*. PESQ and DNSMOS perform much better for they targets at reflecting how the human auditory system perceives noisy speech

data, which could to some extent correlate to how NN models deal with noisy input. But the imperfection suggests that human sensing system is essentially different from NN sensing models and it is necessary to directly evaluate the behavior of NN models.

The experiment results show that existing methods are not designed to evaluate sensing quality in the new context of NN-based machine learning perception algorithms because they do not consider how NN models deal with noise in data. This further motivates us to look at approaches in the machine learning community so as to design a sensing quality measure catered to NN models used for perception. Considering the often non-trivial data collection process, we would also desire an *efficient* measure that can quantify sensing quality with a limited amount of samples.

# 3 SQEE: UNCERTAINTY-BASED SOLUTION

In this section, we elaborate on the design of SQEE, a sensing quality measure in the context of NN-based machine learning perception algorithms requiring only limited data. We adopt the idea of uncertainty measurement, which evaluates the impact of data quality on the inference uncertainty of NN models so as to reflect the sensing quality as perceived by NN models subjectively.

## 3.1 Design of SQEE

Figure 4 shows SQEE's sensing quality evaluation pipeline. Unlike existing methods that evaluate sensing quality independent from the perception model, SQEE evaluates the perception performance of pretrained NN models using state-of-the-art uncertainty measurements. We believe it can better reflect sensing quality in term of machine perception, for a poor data quality ultimately leads the model to be uncertain when making decisions.

The first step of SQEE is to select a subset of examples to be tested. The size of the set depends on the trade-off between the time to spend on the evaluation process and the accuracy we want to achieve. It is desired to have examples drawn from the testing set with the same distribution as the training set of the NN model, to reduce the impact of domain shift. The next step is to decide the candidate locations for the receiver and the transmitter. The receiver location mainly depends on the power type and usage of the sensor, and the set of transmitter location should reflect where sound sources typically are in the real world. To evaluate the sensing quality for one location, the user needs to transmit the sampled testing data at each of the transmitter location. Depending on whether network connection or online pretrained NN model is available, SQEE offers online and offline settings for evaluation. In the online setting, the collected data will be transmitted to a cloud server for evaluation and the runtime bottleneck is the data collection process and network transmission speed. For the offline setting, the collected data will be directly evaluated at the edge device; data collection and the compute capability of edge devices is the major bottleneck in this case. In the context of speech sensing, we consider W2V2 as the cloud model and SEW-tiny as the edge model, as detailed in Section 2.3.

We propose to evaluate sensing quality using the criterion of uncertainty. As an important concept for systems, uncertainty quantifies the phenomena of domain shift and the impact of noise to NN models, thus naturally reflects how data quality affect perception model accuracy. In general, existing uncertainty measurements take as input the predicted probability and estimate uncertainty by analyzing relationships of the predicted probabilities of an ensemble of models. Several measurement variations have been proposed according to the source of uncertainty and some practical considerations of the associated machine learning task. We tested a wide range of variants and chose the best one for SQEE. We view the measured uncertainty score as the sensing quality and rank among all the candidate sensing locations. We next introduce the technical details of uncertainty formulation and the variants we compare with in the following sections.

## 3.2 Uncertainty Formulation

According to Malinin et al. [28], there are two fundamental sources of model uncertainty. The first source is *data uncertainty*, which reflects the input data quality in term of complexity and integrity. The second source is *knowledge uncertainty* due to a lack of knowledge regarding the current input.

Voice sensing is associated with the machine learning task of speech recognition that targets to learn a model that converts input audio signal to a sequence of natural language words of unpredetermined length. This makes speech recognition more complicated than classification or regression, which outputs a single value. Studying voice sensing quality is representative for it can be generalized to other sensing modalities with ease.

We formally describe how the two sources of uncertainty are defined and estimated for structured prediction problem. Let $\boldsymbol{f_\theta}$ denote an NN model that takes as input a sequence of length $T$: $\boldsymbol{x} = \{x_1, x_2, \cdots, x_T\}$ and outputs a sequence of $L$ tokens $\boldsymbol{y} = \{y_1, y_2, \cdots, y_L\}$. $\boldsymbol{\theta}$ is the parameters of $\boldsymbol{f}$. $\boldsymbol{f_\theta}$ works by modeling the conditional probability $\Pr(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ and selecting the maximal as the output.

Bayesian approaches treat model parameters $\boldsymbol{\theta}$ as random variables. By identifying $\mathrm{p}(\boldsymbol{\theta})$, the density function of $\boldsymbol{\theta}$, the true posterior of $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$ can be obtained through Bayes' rule directly. However, the real $\mathrm{p}(\boldsymbol{\theta}$ is intractable for neural networks [29]. The most effective alternative is to construct an ensemble of $M$ speech recognition models $\{\Pr(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ as equivalent samples over the distribution $\mathrm{p}(\boldsymbol{\theta})$. Therefore, the predictive posterior is derived as:

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\theta}}[\Pr(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})] \approx \frac{1}{M}\sum_{m=1}^{M}\Pr(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^{(m)}). \quad (1)$$

The *total uncertainty* is the sum of *data uncertainty* and *knowledge uncertainty*. It can be described by the total 'uncertainty' underlying the predictive posteriors, which can be directly quantified by the conditional entropy:

$$\mathcal{H}[\boldsymbol{y}|\boldsymbol{x}] = \sum_{\boldsymbol{y}} -\Pr(\boldsymbol{y}|\boldsymbol{x})\ln\Pr(\boldsymbol{y}|\boldsymbol{x}). \quad (2)$$

According to the definition of *data uncertainty*, it is the uncertainty underlying input data that exists no matter how the model is trained. So it is defined as the expected model conditional entropy with

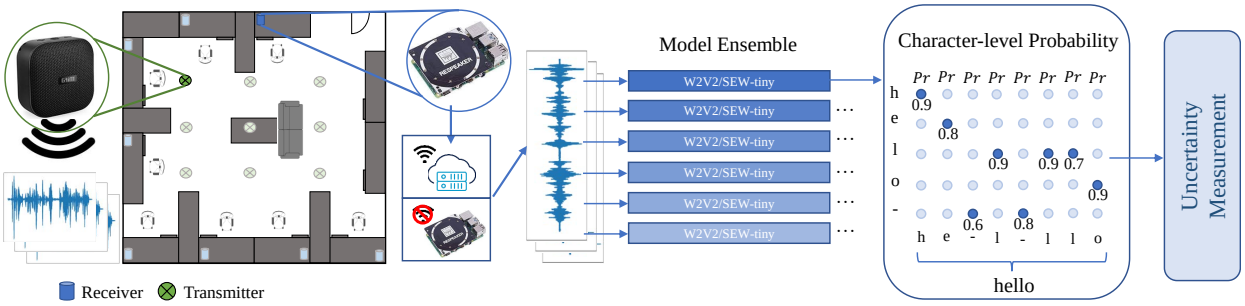



**Figure 4: Overview of SQEE. To evaluate the sensing quality for a sensing location (denoted by the blue cylinder), 1) user deploys the transmitter at each of the transmitting locations (crossed circles); 2) depending on availability of online cloud model, sensing data is transferred to cloud server or kept at the edge. We use W2V2 as the cloud model and SEW-tiny as the edge model; 3) we estimate the quality of sensing location using the estimated uncertainty given the output character-level predicted probability of an ensemble of NN models. After evaluating all the candidate sensing locations, SQEE ranks among them to obtain the best location.**

regard to $\boldsymbol{\theta}$:

$$\mathbb{E}_{\boldsymbol{\theta}}[\hat{\mathcal{H}}[\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}]] \approx \frac{1}{M}\sum_{m=1}^{M}\sum_{\boldsymbol{y}} -\Pr(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^{(m)})\ln\Pr(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^{(m)}). \tag{3}$$

For *knowledge uncertainty*, it measures the discrepancy between the training data (knowledge) and the prediction from the model. As the learnt knowledge is represented by the model parameter $\boldsymbol{\theta}$, *knowledge uncertainty* is defined as the mutual information between $\boldsymbol{y}$ and $\boldsymbol{\theta}$:

$$\mathcal{I}[\boldsymbol{y};\boldsymbol{\theta}|\boldsymbol{x}] = \mathcal{H}[\boldsymbol{y}|\boldsymbol{x}] - \mathbb{E}_{\boldsymbol{\theta}}[\hat{\mathcal{H}}[\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}]]. \tag{4}$$

In the machine learning community, *knowledge uncertainty* draws more attention from researchers because it can be reduced if the model is provided with more training data, while *data uncertainty* is considered irreducible because it is a property underlying the associated data distribution. However, in the context of our sensing quality evaluation, *data uncertainty* is the major source of uncertainty caused by the variances intruded during signal propagation. *Knowledge uncertainty* also exists. Although the transmitted clean signal is generated from the same distribution as the training set of the model, the received signal is interpolated with noise, leading to change of distribution.

## 3.3 Uncertainty Estimation for Speech Recognition

We next discuss how uncertainty is estimated for speech recognition models in practice.

*3.3.1 CTC-based Uncertainty.* Specifically, the pretrained models used for benchmark evaluation utilizes a CTC [12] setting to generate output sequence. CTC is the most common framework used for training a speech recognition model. An advantage of CTC is that it does not require temporal alignment between the input speech audio and the target sequence, but marginalizes among all possible alignments allowed by the algorithm. As a result, the basic assumption of CTC models made is the conditional independence between any predicted tokens $y_i$ and $y_j$. A speech recognition model is trained to model $\Pr(y_i|\boldsymbol{x})$, $i = 1$ to $L$. $L$ is a predefined max target sequence length and $y_i$ represents the basic characters of the spoken language. To recover the spoken words, a separate decoding algorithm is used to merge consecutive identical characters. Under the independent assumption, one can easily compute the probability of any character-sequence given input speech signal via $\Pr(\boldsymbol{y}|\boldsymbol{x}) = \prod_i \Pr(y_i|\boldsymbol{x})$. In this way, we can apply chain rule to compute the three uncertainty measurements. Take the *total uncertainty* as an example, Eq. (2) can be further decomposed as

$$\mathcal{H}[\boldsymbol{y}|\boldsymbol{x}] = \frac{1}{L}\sum_{i=1}^{L}\sum_{y_i} -\Pr(y_i|\boldsymbol{x})\ln\Pr(y_i|\boldsymbol{x}), \tag{5}$$

where $\Pr(y_i|\boldsymbol{x})$ can be approximated similarly as Eq. (1). Following Eq. (5), *data uncertainty* and *knowledge uncertainty* can be decomposed the same way.

*3.3.2 CTC-LM-based Uncertainty.* The assumption of conditional independence made by CTC models is too strong for real natural languages. Instead, there exist contextual rules that help speech recognition models calibrate the generated text. Language modeling (LM) is a popular technique used in the field of Natural Language Processing, which explicitly models the conditional probability of the next token given the previous ones $\Pr(y_i|\boldsymbol{x},\boldsymbol{y}_{1:i-1})$ [5]. Typically, n-gram LM considers only the previous $n$ tokens due to the fact that closer tokens are more important to the next token than tokens exist earlier, thus $\Pr(y_i|\boldsymbol{x},\boldsymbol{y}_{1:i-1}) = \Pr(y_i|\boldsymbol{x},\boldsymbol{y}_{i-n:i-1})$. It has been verified that character-level LM can be combined with CTC speech recognition model at the bottom, which can further improve the performance of speech sensing [44].

In general, CTC-LM-based models learn the conditional probability $\Pr(y_i|\boldsymbol{x},\boldsymbol{y}_{i-n:i-1})$. Due to the n-gram dependency, deriving $\max_{\boldsymbol{y}}\Pr(\boldsymbol{y}|\boldsymbol{x})$ is computationally impractical because one has to to exhaustively compute the probability $\Pr(\boldsymbol{y}|\boldsymbol{x})$ for each of the possible output sequence $\boldsymbol{y}$. A common solution to CTC-LM-based model is beam search [30], which maintains a max heap of a fixed size $K$ storing the most likely $K$ sequences for each step $i = 1$ to $L$. For each step, it only compares possible sequences generated from the stored $K$ sequences to save time and space. By viewing beam search as a process of importance sampling, the *data uncertainty* of

a CTC-LM-based speech recognition model can be approximated as:

$$\mathcal{H}[\boldsymbol{y}|\boldsymbol{x}] \approx \frac{1}{ML}\sum_{m=1}^{M}\sum_{k=1}^{K}\pi_k\sum_{i=1}^{L}\mathcal{E}(\Pr(y_i^{(k)}|\boldsymbol{x},\boldsymbol{y}_{1:i-1}^{(k)},\boldsymbol{\theta}^{(m)})),$$

where $\boldsymbol{y}^{(k)}, k = 1$ to $K$ are the $K$ samples given by the beam search algorithm, $\pi_k \propto \Pr(\boldsymbol{y}^{(k)}|\boldsymbol{x})$ subject to $\sum_{k=1}^{K}\pi_k = 1$ is the weight associated with the samples and $\mathcal{E}(p)$ represents the entropy term for a single probability value. However, the overall predictive posterior of one single token $\Pr(y_i|\boldsymbol{x})$ cannot be computed using Eq. (1) for $y_i$ depends on previous predictions.

In practice, the performance of token-level CTC-LM is unsatisfactory because the ambiguity underlying CTC decoding process where different n-gram tokens can lead to the same decoded words. Word-level CTC-LM is mostly used instead. However, the ambiguity of CTC decoding also exists, which makes it non-trivial to compute the word-level conditional probability $\Pr(z_i|\boldsymbol{x}, z_{1:i-1}, \boldsymbol{\theta}^{(m)})$ for word $z_i$. As a result, in this paper, we use sequence-level entropy, rather than token-level entropy to estimate *data uncertainty* for word-level CTC-LM models, which is derived as:

$$\mathcal{H}[\boldsymbol{y}|\boldsymbol{x}] \approx \frac{1}{ML}\sum_{m=1}^{M}\sum_{k=1}^{K}\pi_k\mathcal{E}(\Pr(\boldsymbol{z}^{(k)}|\boldsymbol{x},\boldsymbol{\theta}^{(m)})). \tag{6}$$

Existing speech recognition models can be categorized by the conditional dependency on previously generated tokens. To generalize to other NN models which assume the conditional dependency, one can follow CTC-LM-based uncertainty, i.e., Eq. (6); for the NN models which assume the conditional independence, one can follow the CTC-based uncertainty, i.e., Eq. (5).

*3.3.3 Output Uncertainty.* Both CTC-based and CTC-LM-based model are not fully end-to-end models for it requires a decoding algorithm to recover the output sequence from the character-level probability. As a result, the uncertainty underlying the predicted probability does not necessarily correlate with the uncertainty underlying the final output sequence. As it is impossible to quantify the probability for the decoded output, edit distance is used to heuristically estimate the *output uncertainty* of the ensembles of CTC-based models [47].

Let $\boldsymbol{z}^{\boldsymbol{i}}$, $i = 1$ to $M$ denote the output sequences of the $M$ ensembles of models. The difference between $\boldsymbol{z}^{\boldsymbol{i}}$ and $\boldsymbol{z}^{\boldsymbol{j}}$ is captured by the word-level edit distance $E_{ij}$. The *output uncertainty* is estimated through the average output variations as $\frac{\sum_i\sum_j E_ij}{\sum_i|\boldsymbol{z}^{\boldsymbol{i}}|}$.

## 3.4 Model Ensemble

After obtaining the approximation of the sources of uncertainty, we discuss how the model ensemble, i.e., the $M$ model weights $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)} \cdots, \boldsymbol{\theta}^{(M)}$ are generated. The most promising model ensemble approaches are Deep Ensembles and MC-dropout. Deep Ensembles generate each of them using random weight initialization accompanied with randomly shuffled training data. However, Deep Ensembles is not applicable to large-scale NN model like W2V2 for it is overwhelmingly time consuming to train the ensemble of model weights. MC-dropout is a more desirable ensemble approach for it does not require model training and it is very computationally cheap, and therefore fits well deployment on edge devices. Conventionally, MC-dropout applies random weight dropout to all the layers of the NN model by randomly masking the hidden vectors with 0 in a given probability $p$. All-layer MC-dropout can be further simplified with dropout only at the last layer. Both all-layer and last-layer dropout have been empirically verified to be effective approximation of Bayesian NNs, by which the constructed model ensemble well estimates model uncertainty. As a result, both of the implementations of MC-dropout are studied in this work.

# 4 EVALUATION OF SQEE

## 4.1 Implementation Details

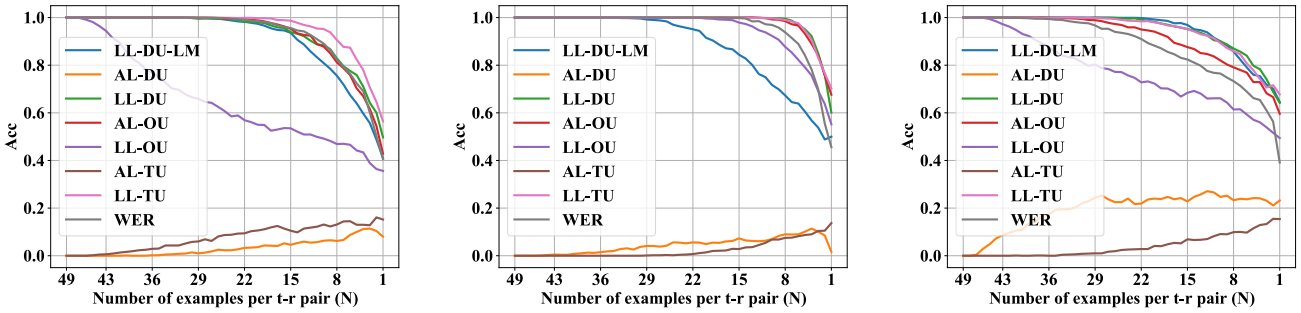
We introduce the implementation details of the uncertainty measurement variants we compare with. There are two variations of MC-dropout used for model ensemble: all-layer dropout that applies to all the NN weights and last-layer dropout that applies to only the last fully connected layer. We use prefix **AL-** to denote all-layer dropout and **LL-** to denote last-layer dropout. The ensemble size $M$ is 10 and the dropout probability $p$ is set to 0.01 and 0.9 for **AL-** and **LL-** respectively. We choose very low dropout probability for **AL-** because the dropout effect is exponentially enhanced based on the number of NN layers.

We compare the use of CTC scheme and CTC-LM scheme with W2V2 model and SEW-tiny model. For the CTC scheme, the prediction is independent among all time steps, and thus we can derive the overall entropy directly. However, for the CTC-LM scheme, the prediction of current time step depends on the previous predictions and it is impractical to exhaust over the vast target space. So we use beam search to approximate weighted sampling. We use postfix **-LM** to denote that we use CTC-LM with beam search and *no suffix* implies that we use CTC. For CTC-LM, the beam search size is 10 and we preserve the top $K = 10$ predictions.

For uncertainty measurements, since we mainly focus on the impact of data quality on inference performance, *knowledge uncertainty* is a minor aspect of uncertainty source. As a result, we experiment on the performance of *total uncertainty*, *data uncertainty*, *output uncertainty* and use **TU**, **DU** and **OU** to refer to the three types of uncertainty respectively. As we argued in Section 3.3.2, we only tested *data uncertainty* (**DU**) for CTC-LM.

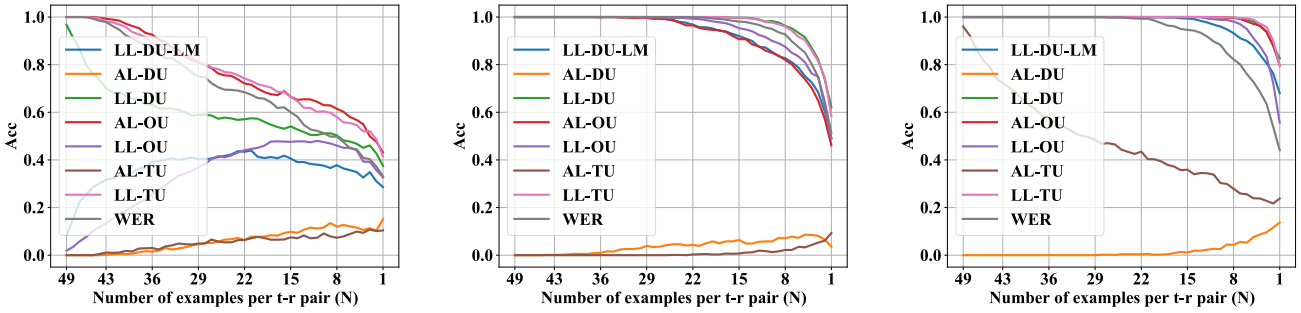
## 4.2 Experiment Results

The performance of all the compared variants for the edge model SEW-tiny and the cloud model W2V2 is shown in Figure 5 and Figure 6. In general, **LL-TU** and **LL-DU** are the best measures for sensing quality evaluation, among which **LL-TU** is slightly better for it significantly outperforms **LL-DU** on *office* for W2V2 model. We notice that three locations have similar sensing quality with respect to W2V2 model on *office*, which suggests that the *knowledge uncertainty* term underlying **TU** can help models distinguish sensing quality at a finer resolution. **LL-DU-LM** underperforms WER in most of the environments. The extra knowledge introduced by LM helps calibrate the prediction probability by considering the contextual dependencies, however harms sensing quality measurement. It is because extra knowledge weakens the impact of data quality on the entropy of predicted probabilities, making it difficult to distinguish good sensing location and bad sensing location. The




(a) Office (b) Bedroom (c) Living room

**Figure 5: Offline Uncertainty Measure Comparison. We visualize the top-1 accuracy when 1000 random sets of $N$ examples are tested using edge model SEW-tiny.**




(a) Office (b) Bedroom (c) Living room

**Figure 6: Online Uncertainty Measure Comparison. We visualize the top-1 accuracy when 1000 random sets of $N$ examples are tested using cloud model W2V2.**

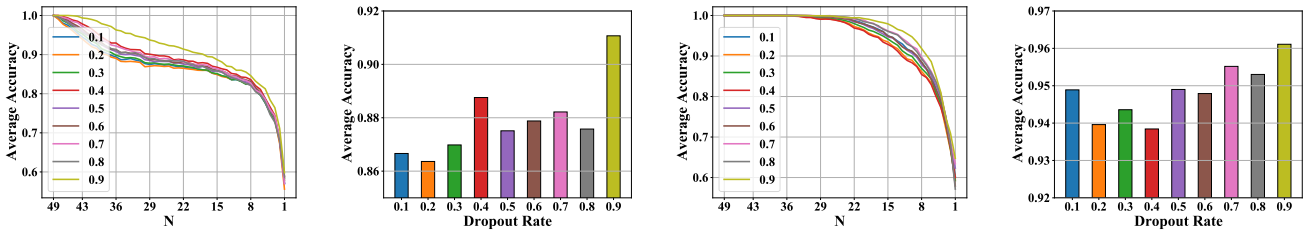
performance of **LL-OU** is not consistently good and it even fails to find the best location when all the data is available for W2V2 on *office*, which shows that its estimation is coarse-grained. Compared with **LL-OU**, **AL-OU** is much better. The result demonstrates that all layer MC-dropout is good at generating random output while maintaining the output quality.

Speech sensing NN models implemented in CTC scheme learns to model the token level probability. To obtain the final prediction, the models require decoding algorithms to eliminate redundant tokens and merge them to real words. It is possible that different probability predictions have the same decoded sequence. However, we cannot exhaustively iterate all the token sequences to compute word sequence probability. Therefore, all the measures like **DU-LM** and **WER** that evaluate sensing quality using the final word-level output actually introduce knowledge from the decoding algorithm, making them unable to distinguish between locations that have similar sensing quality.

Although all-layer MC-dropout is a better approximation to Bayesian NNs, *total uncertainty* and *data uncertainty* with all-layer dropout can not perform well: in most of the cases, these all-layer dropout approaches cannot find the best location correctly. We argue that all-layer dropout is too strong to large-scale deep pre-trained NN models and the 0.01 dropout probability we choose is not an optimal choice but a compromise. Since the dropout effect is exponentially enhanced with the number of NN layers, it is very difficult to tune the dropout rate: very small dropout rate may not introduce necessary randomness to the model, whereas slightly higher dropout rate might destroy all the forward-pass paths, making the model fail to output anything meaningful. In our experiment, the 0.01 dropout rate is a good choice for estimation of *output uncertainty* using edit distance but still too large to other uncertainty measures based on the predicted probability. As a result, all layer dropout method is not a desired strategy for NN models for one needs to exhaustively find the best all layer dropout rate for one environment, and needs to repeat the whole process again upon environmental change.

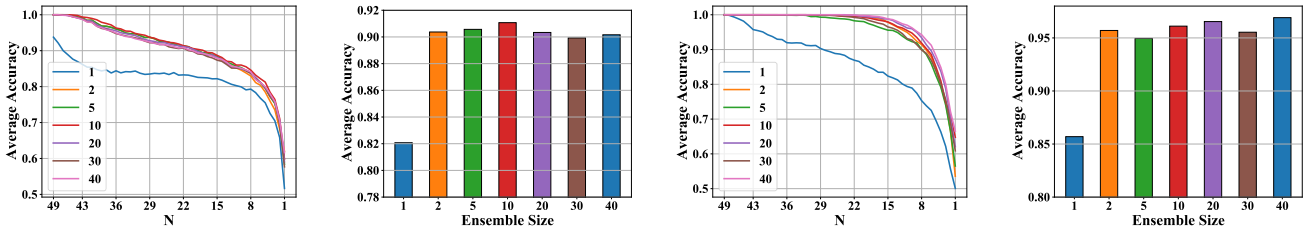
## 4.3 Dropout Rate

Last-layer dropout has demonstrated good performance and better fits pretrained deep NN models. To further automate the sensing quality evaluation pipeline, we experiment on different last-layer dropout rates using grid search to study the criterion on deciding the dropout rate. We focus on **LL-TU**, the best method according to the previous experiments. As shown in Figure 7, we visualize the averaged accuracy of all the three environments v.s. the number of testing data for W2V2 and SEW-tiny in (a) and (c). Shown in (b) and (d), we also visualize the area under curve (AUC) in (a) and (c) correspondingly, which we view as the averaged performance for different dropout rate.




(a) W2V2 (b) W2V2 (c) SEW-tiny (d) SEW-tiny

**Figure 7: Comparison of different dropout rates for LL-TU measure. Figure (a) and (c) are the curves of averaged accuracy v.s. number of testing data. Figure (b) and (d) visualizes the area under curve in (a) and (c) correspondingly.**




(a) W2V2 (b) W2V2 (c) SEW-tiny (d) SEW-tiny

**Figure 8: Comparison of different ensemble sizes for LL-TU measure. Figure (a) and (c) are the curves of averaged accuracy v.s. number of testing data. Figure (b) and (d) visualizes the area under curve in (a) and (c) correspondingly.**

In general, the performance improves with the dropout rate increasing. From (a) and (c), we observe that the performance of different dropout rate mainly follows the same trend with the number of testing data $N$ descending, and the best dropout rate curve with the largest enclosed area is $p = 0.9$, which also stands out clearly in (b) and (d). This experiment shows that it is unnecessary to tune the dropout rate for **LL-TU** method for a higher dropout rate is desired for it guarantees adequate randomness within ensemble of NN models. This is because both W2V2 and SEW-tiny are are usually trained with dropout enabled so they are typically robust. In practice, in order to implement SQEE on a new sensing system using different pretrained NN models, one just needs to set the dropout at 0.9 for the models usually have large amount of parameters as well. Alternatively, one can conduct further experiments using some offline data to decide the optimal dropout rate if needed when the size of the model is much smaller.

## 4.4 Number of Ensemble Models

We inspect the effect of number of ensemble models on the chosen **LL-TU** method. In Figure 8, the averaged accuracy of all the three environments for W2V2 and SEW-tiny are shown in (a) and (c) and the averaged performance is shown in (b) and (d). We observe a significant drop in performance when there is no ensemble models, suggesting that we need at least two samples from the model weight distribution $\Pr(\theta)$ to reduce bias in estimating the predictive posterior in Eq. (1). Theoretically, uncertainty measurement performance should improve as ensemble size increasing for we can estimate predictive posterior more accurately. However, such improvement is not evident in the experiment: we can roughly detect an improving trend for SEW-tiny but not for W2V2. The experiment results further suggest that we only need ensemble of two models for the offline deployment, where the CPU inference time is one of the bottleneck and we can use much larger ensemble size for the online deployment to achieve better performance.

## 4.5 Real-world Performance Analysis

After verifying the performance of uncertainty measurement variants and the choices of key hyperparameters, we deploy SQEE illustrated in Section 3.1 in the real world. We use **LL-TU** to estimate sensing quality, where the dropout rate is set to 0.9 and we use 10-model ensemble for the cloud deployment and 2-model ensemble for the edge deployment. We deploy the evaluation algorithm in the same ReSpeaker used for dataset collection. The runtime environment at the edge is on the associated Raspberry Pi with an ARM Cortex-A72 CPU, 4 GB RAM and 8 GB swap memory. The cloud runtime environment is on a GPU server with RTX A6000.

We evaluate the performance of SQEE in the real world environment and focus on the top-1 accuracy, top-2/top-3 accuracy and the runtime. The real-world performance of the edge deployment is shown in Table 2 and the performance of the cloud deployment is shown in 3. We compare the top-3 accuracy in *office* and *bedroom* for there are 8 candidate sensing locations and we compare the top-2 accuracy in *living room* for there are only 6 locations. We also compare the estimated runtime for each t-r pair. For the offline deployment, the runtime is calculated as the time for collecting $N$ examples for all the t-r pairs added with the time used for quality evaluation and then averaged over the number of t-r pairs. Both collection and evaluation are the major source of runtime. When deployed on the cloud, it is estimated by adding the data collection time, data transfer time and online evaluation time and then averaging; data collection time is the major source of runtime.

**Table 2: Offline Runtime Performance for SEW-tiny Model. We compare the top-1 accuracy, top-3 accuracy and the averaged running time per t-r pair for *office* and *bedroom*. We compare the top-2 accuracy instead of top-3 accuracy for *living room* since there are fewer candidate sensing locations.**

| *office* | Top-1 Acc | Top-3 Acc | T-R time (s) |
|---|---|---|---|
| N = 1 | 0.490 | 0.835 | 22.79 |
| N = 2 | 0.604 | 0.924 | 40.41 |
| N = 3 | 0.683 | 0.961 | 55.22 |
| N = 4 | 0.722 | 0.977 | 70.76 |
| N = 5 | 0.778 | 0.989 | 83.40 |
| N = 10 | 0.891 | 1.000 | 158.70 |

| *bedroom* | Top-1 Acc | Top-3 Acc | T-R time (s) |
|---|---|---|---|
| N = 1 | 0.557 | 0.831 | 24.43 |
| N = 2 | 0.694 | 0.870 | 37.65 |
| N = 3 | 0.798 | 0.931 | 54.82 |
| N = 4 | 0.867 | 0.961 | 75.13 |
| N = 5 | 0.912 | 0.980 | 87.97 |
| N = 10 | 0.982 | 0.999 | 164.06 |

| *living room* | Top-1 Acc | Top-2 Acc | T-R time (s) |
|---|---|---|---|
| N = 1 | 0.560 | 0.857 | 23.78 |
| N = 2 | 0.700 | 0.944 | 42.42 |
| N = 3 | 0.765 | 0.975 | 58.98 |
| N = 4 | 0.795 | 0.997 | 69.65 |
| N = 5 | 0.828 | 0.997 | 87.39 |
| N = 10 | 0.932 | 1.000 | 170.24 |

**Table 3: Online Runtime Performance for W2V2 Model. We compare the top-1 accuracy, top-3 accuracy and the averaged running time per t-r pair for *office* and *bedroom*. We compare the top-2 accuracy instead of top-3 accuracy for *living room* since there are fewer candidate sensing locations.**

| *office* | Top-1 Acc | Top-3 Acc | T-R time (s) |
|---|---|---|---|
| N = 1 | 0.414 | 0.961 | 15.92 |
| N = 2 | 0.486 | 0.972 | 25.83 |
| N = 3 | 0.521 | 0.991 | 32.27 |
| N = 4 | 0.520 | 0.998 | 45.51 |
| N = 5 | 0.546 | 0.998 | 53.85 |
| N = 10 | 0.604 | 1.000 | 87.69 |

| *bedroom* | Top-1 Acc | Top-3 Acc | T-R time (s) |
|---|---|---|---|
| N = 1 | 0.584 | 0.887 | 15.52 |
| N = 2 | 0.746 | 0.958 | 26.03 |
| N = 3 | 0.814 | 0.982 | 34.60 |
| N = 4 | 0.861 | 0.996 | 42.62 |
| N = 5 | 0.903 | 1.000 | 55.54 |
| N = 10 | 0.977 | 1.000 | 91.50 |

| *living room* | Top-1 Acc | Top-2 Acc | T-R time (s) |
|---|---|---|---|
| N = 1 | 0.796 | 0.937 | 17.06 |
| N = 2 | 0.902 | 0.983 | 25.15 |
| N = 3 | 0.958 | 0.999 | 33.15 |
| N = 4 | 0.972 | 1.000 | 46.43 |
| N = 5 | 0.989 | 1.000 | 54.83 |
| N = 10 | 0.999 | 1.000 | 95.13 |

In general, the top-1 accuracy for both deployments in all the environments can reach around $90\%$ when $N = 10$ examples are tested for each t-r pair except for the online deployment for *office* due to that there are $3$ locations with similar sensing quality for W2V2 shown in Figure 2, which suggests that $N = 10$ examples are needed to reach the best evaluation performance. However, it takes more than $2.5$ and $1.5$ minutes to finish the evaluation of each t-r pair for the edge and cloud deployment respectively, which is not very promising for household use. To trade-off between runtime and performance, we argue that only $N = 2$ examples are needed for each t-r pair, and the top-3 accuracy can also reach more than $90\%$ on average. In this case, it only takes around $40$ seconds to test one t-r pair offline, and the overall runtime is less that $50$ minutes for *office* that has the most t-r pairs. We also note that the shown runtime can be further accelerated by $40\%$ by simply enabling the procedures of data collection and data evaluation to run in parallel. Therefore, the offline runtime can be reduced to nearly the same as the online runtime where the only bottleneck is data collection which cannot be further optimized anymore.

## 5 GENERALIZATION DISCUSSION

We discuss how to generalize SQEE to other modalities and scenarios using WiFi CSI-based sensing as an example.

This paper aims to identify the best location to deploy sensors in a new environment so that a trained NN model can achieve the best performance in this new environment. The only assumption in SQEE is that the trained NN model to be deployed should have reasonable performance in this new environment. Therefore, SQEE can be applied to most of the existing sensing modalities including camera-bases sensing, RF sensing, and vibration-based sensing, as NN models' generalization capability have been verified [20, 24, 40]. Please note that different evaluation metrics for NN models other than WER will not affect the generalization of SQEE, as long as the NN models were optimized for the same metric(s).

Let's now take WiFi-CSI based human activity sensing as an example to illustrate generalization of SQEE. The task here is to classify human subject's activity given WiFi-CSI data transmitted from a transmitter and collected from multiple receivers. Formally speaking, given input WiFi CSI $\boldsymbol{X}$, the NN model predicts the probability of each activity $\Pr(y|\boldsymbol{X}, \boldsymbol{\theta})$. The ground-truth sensing quality is defined as the classification accuracy of all the CSI data collected, and it is associated with the deployment location combination of the transmitter and receivers. As a classification problem, one can estimate the predictive posterior of model ensemble following Eq. (1) and then calculate *total uncertainty* following Eq. (2). To evaluate the sensing quality for each deployment, one can use last-layer MC-dropout as it can be efficiently implemented and *total uncertainty* to capture all the discrepancy between training and testing data. Since the deployment here involves multiple receivers, depending on the search space, one can further use exhaustive search or approximation methods like simulated annealing or genetic algorithm to find/approximate the best deployment.

## 6 RELATED WORK

Our work studies sensing quality from a machine perception perspective, using speech data as an example. It is generally related to traditional audio signal quality evaluation metrics and recent works on uncertainty measurement for NN models. It is also related to IoT data quality evaluation, but from a different point of view that focuses on the impact on machine learning model performance.

### 6.1 Audio Signal Quality Evaluation

From the view of signal processing, distortion is considered as a straightforward metric for audio signal quality evaluation. A typical set of works focus on estimating SNR [15, 21, 33, 34]. SSIM [17] measures correlation between degraded and perceived signals and STOI [45] measures the correlations between temporal envelopes. However, noise may not necessarily prompt to degraded hearing quality because well-shaped distorting noise in audio signals can be inaudible due to property of human auditory systems [46].

Consequently, subjective methods are proposed to directly evaluate human perception quality. The most common framework is ITU BS.1284 [35]. As it is usually impractical to perform a formal subjective test, a wide range of works focus on developing objective metrics with perceptual models that incorporate properties of human auditory systems. PEAQ [46], PESQ [36] and POLQA [37] are a series of ITU recommended algorithm. These works are rule-based intrusive measurement that requires full reference of clean signals. Recently, NN models are further used to estimate human MOS from a non-intrusive manner [2, 38].

Although standard audio quality metrics have been deeply studied and widely applied in the past decades, audio quality evaluation algorithm in term of machine perception is understudied.

### 6.2 Neural Networks Uncertainty Evaluation

Predictive uncertainty is crucial to deployment of NN models under data distribution shift and out-of-distribution (OOD) inputs [1]. A majority of prior works follow a Bayesian formalism [6], approximate Bayesian NNs [8, 14] and quantify the uncertainty as the posterior distribution given the training data. But Bayesian NNs are hard to implement and computationally inefficient [23]. Ensemble NNs, such as MC-dropout [10] and Deep Ensembles [23] are experimentally verified as strong alternatives to Bayesian-based methods that can be efficiently implemented. Furthermore, researchers decompose overall uncertainty into data uncertainty and knowledge uncertainty and found knowledge uncertainty useful for detecting OOD inputs and classification errors [28, 41].

Most of the previous works focus on unstructured tasks, where typical takes are image classification and time series regression. Efforts are made to further extend the scope to estimation for structured tasks like machine translation and speech recognition. Wang et al. [48] and Xiao et al. [50] apply beam search [11] and MC-dropout to machine translation. However, their solutions cannot pinpoint the source of uncertainty. Ott et al. [31] generalize uncertainty decomposition into machine translation. Malinin et al. [29] extend ensemble-based approaches to both machine translation and speech recognition tasks and solidly derive estimation of uncertainty terms. Their work extensively studied multiple choices of methods on the tasks of error detection and OOD detection.

Build upon recent works, this paper is the first one attempt to use uncertainty to study the quality of sensing data.

### 6.3 IoT Data Quality

Karkouch et al. comprehensively study the data quality issue in IoT systems [18]. They evaluate data quality from 5 dimensions including timeliness, completeness, accuracy, confidence and data volume. The data quality dimensions are further extended to 4 broader categories, intrinsic feature, accessibility, contextual integrity and representation [9]. On the other hand, Banerjee et al. narrow down the scope to different application needs from a computer system perspective [4]. These works are mainly defining qualitative sensing quality dimensions based on a general IoT system, lacking a benchmark for quantitative evaluation of sensing quality [42]. They do not consider a machine learning perception model in the IoT system as well.

Recently, AutoQual evaluates vibration-based sensing quality with respect to machine learning model performance[53]. However, AutoQual is designed for structural vibration sensing and requires domain knowledge to specify environmental impacts on data for different sensing modalities, which makes it difficult to generalize across IoT systems, e.g. speech recognition. Compared with AutoQual, our work targets at measuring sensing quality for sophisticated pretrained NN models; we also discussed generalization of our method to other modalities in detail.

## 7 CONCLUSIONS

This work proposed an efficient sensing quality measure with limited data samples from the view of machine perception. To benchmark the evaluation pipeline, we collected data in three different indoor environments from more than 140 t-r pairs. Existing signal quality measures are independent from the associated sensing task, thus failing to reflect the impact on the machine learning models used for perception. We explored the use of uncertainty measurement for sensing quality evaluation. We proposed the framework of SQEE, which involves three steps of sensing data collection, transmission and evaluation and supports both online and offline settings. We chose **LL-TU** that estimates *total uncertainty* using last layer MC-dropout for it is the best uncertainty measurement among a wide range of variations after exhaustive comparisons. We further experimented on different choices of dropout rate and ensemble size. We implemented SQEE on edge devices and evaluated its efficacy by comparing the trade-off among top-1 accuracy, top-2/top-3 accuracy and the actual running time. In general, SQEE only requires $N = 2$ samples for each t-r pair to reach descent top-2/top-3 performance.

For future directions, we plan to extend SQEE to other sensing modalities. We also plan to explore situations where there are multiple sensing tasks associated with the sensing data, and scenarios where pretrained models are not available.

## ACKNOWLEDGMENTS

This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. It was also supported by Halıcıoğlu Data Science Institute. This work was also supported in part by National Science Foundation Convergence Accelerator under award OIA-2040727 as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

# REFERENCES


[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* (2016).

[2] Anderson R Avila, Hannes Gamper, Chandan Reddy, Ross Cutler, Ivan Tashev, and Johannes Gehrke. 2019. Non-intrusive speech quality assessment using neural networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 631–635.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems* 33 (2020), 12449–12460.

[4] Tanvi Banerjee and Amit Sheth. 2017. Iot quality control for data and application needs. *IEEE Intelligent Systems* 32, 2 (2017), 68–73.

[5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in Neural Information Processing Systems* 13 (2000).

[6] José M Bernardo and Adrian FM Smith. 2009. *Bayesian theory*. Vol. 405. John Wiley & Sons.

[7] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. 2008. *Guide to the Expression of Uncertainty in Measurement*. JCGM 100:2008, GUM 1995 with minor corrections.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*. PMLR, 1613–1622.

[9] John Byabazaire, Gregory O'Hare, and Declan Delaney. 2020. Data quality and trust: A perception from shared data in IoT. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 1–6.

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

[11] Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711* (2012).

[12] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.

[13] Tatsuhito Hasegawa. 2020. Smartphone sensor-based human activity recognition robust to different sampling rates. *IEEE Sensors Journal* 21, 5 (2020), 6930–6941.

[14] José Miguel Hernández-Lobato and Ryan Adams. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*. PMLR, 1861–1869.

[15] Hans-Günter Hirsch and Christoph Ehrlicher. 1995. Noise estimation techniques for robust speech recognition. In *1995 International conference on acoustics, speech, and signal processing*, Vol. 1. IEEE, 153–156.

[16] Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani Srivastava. 2019. SenseHAR: a robust virtual activity sensor for smartphones and wearables. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 15–28.

[17] Srivatsan Kandadai, Joseph Hardin, and Charles D Creusere. 2008. Audio quality assessment using the mean structural similarity measure. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 221–224.

[18] Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. 2016. Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications* 73 (2016), 57–81.

[19] Panagiotis Kasnesis, Charalampos Z Patrikakis, and Iakovos S Venieris. 2018. PerceptionNet: a deep convolutional neural network for late sensor fusion. In *Proceedings of SAI Intelligent Systems Conference*. Springer, 101–119.

[20] Asif Khan, Dae-Kwan Ko, Soo Chul Lim, and Heung Soo Kim. 2019. Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network. *Composites Part B: Engineering* 161 (2019), 586–594.

[21] Chanwoo Kim and Richard M Stern. 2008. Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In *Ninth Annual Conference of the International Speech Communication Association*.

[22] Katy Klauenberg, Gerd Wübbeler, Bodo Mickan, Peter Harris, and Clemens Elster. 2015. A tutorial on Bayesian normal linear regression. *Metrologia* 52, 6 (2015), 878.

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

[24] Guohao Lan, Mohammadreza F Imani, Zida Liu, José Manjarrés, Wenjun Hu, Andrew S Lan, David R Smith, and Maria Gorlatova. 2021. MetaSense: Boosting RF sensing accuracy using dynamic metasurface antenna. *IEEE Internet of Things Journal* 8, 18 (2021), 14110–14126.

[25] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. 2021. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 286–293.

[26] Zhong Qiu Lin, Audrey G Chung, and Alexander Wong. 2018. Edgespeechnets: Highly efficient deep neural networks for speech recognition on the edge. *arXiv preprint arXiv:1810.08559* (2018).

[27] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenyao Xu, and Kui Ren. 2021. Wavoice: A Noise-resistant Multi-modal Speech Recognition System Fusing mmWave and Audio Signals. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 97–110.

[28] Andrey Malinin. 2019. *Uncertainty estimation in deep learning with application to spoken language assessment*. Ph. D. Dissertation. University of Cambridge.

[29] Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650* (2020).

[30] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al. 1977. Speech understanding systems: Report of a steering committee. *Artificial Intelligence* 9, 3 (1977), 307–316.

[31] Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*. PMLR, 3956–3965.

[32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.

[33] Cyril Plapous, Claude Marro, and Pascal Scalart. 2006. Improved signal-to-noise ratio estimation for speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 6 (2006), 2098–2108.

[34] Lawrence R Rabiner and Bernard Gold. 1975. Theory and application of digital signal processing. *Englewood Cliffs: Prentice-Hall* (1975).

[35] ITU-R Rec.BS.1284-2. 2019. General Methods for the Subjective Assessment of Sound Quality. *International Telecommunication Union* (2019).

[36] ITU-T Rec.P.862. 2001. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-end Speech Quality Assessment of Narrow-band Telephone Networks and Speech Codecs. *International Telecommunication Union* (2001).

[37] ITU-T Rec.P.863. 2018. Perceptual Objective Listening Quality Prediction. *International Telecommunication Union* (2018).

[38] Chandan KA Reddy, Vishak Gopal, and Ross Cutler. 2021. DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6493–6497.

[39] Yuan Shangguan, Jian Li, Qiao Liang, Raziel Alvarez, and Ian McGraw. 2019. Optimizing speech recognition for the edge. *arXiv preprint arXiv:1909.12408* (2019).

[40] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[41] Lewis Smith and Yarin Gal. 2018. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533* (2018).

[42] Shaoxu Song and Aoqian Zhang. 2020. IoT data quality. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3517–3518.

[43] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2828–2837.

[44] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. 2019. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460* (2019).

[45] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2011. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 7 (2011), 2125–2136.

[46] Thilo Thiede, William C Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G Beerends, and Catherine Colomes. 2000. PEAQ-The ITU standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society* 48, 1/2 (2000), 3–29.

[47] Apoorv Vyas, Pranay Dighe, Sibo Tong, and Hervé Bourlard. 2019. Analyzing uncertainties in speech recognition using dropout. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6730–6734.

[48] Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. *arXiv preprint arXiv:1909.00157* (2019).

[49] Felix Wu, Kwangyoun Kim, Jing Pan, Kyu Han, Kilian Q Weinberger, and Yoav Artzi. 2021. Performance-Efficiency Trade-offs in Unsupervised Pre-training for Speech Recognition. *arXiv preprint arXiv:2109.06870* (2021).

[50] Tim Z Xiao, Aidan N Gomez, and Yarin Gal. 2019. Wat heb je gezegd? detecting out-of-distribution translations with variational transformers. In *Bayesian Deep Learning Workshop (NeurIPS)*.

[51] Shichao Xu, Yangyang Fu, Yixuan Wang, Zheng O'Neill, and Qi Zhu. 2021. Learning-based framework for sensor fault-tolerant building HVAC control with model-assisted learning. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 1–10.

[52] Shuochao Yao, Ailing Piao, Wenjun Jiang, Yiran Zhao, Huajie Shao, Shengzhong Liu, Dongxin Liu, Jinyang Li, Tianshi Wang, Shaohan Hu, et al. 2019. Stfnets: Learning sensing signals from the time-frequency perspective with short-time fourier neural networks. In *The World Wide Web Conference*. 2192–2202.

[53] Yue Zhang, Zhizhang Hu, Susu Xu, and Shijia Pan. 2021. AutoQual: task-oriented structural vibration sensing quality assessment leveraging co-located mobile sensing context. *CCF Transactions on Pervasive Computing and Interaction* 3, 4 (2021), 378–396.

[54] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. 2020. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504* (2020).