Universität
Augsburg
University

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF AUGSBURG

Bachelor’s Thesis in Bachelor Informatik
at Chair of Embedded Intelligence for Health Care and
Wellbeing

Calibrated and Uncertainty-aware Multimodal Emotion Recognition

Anonymous

Universität
Augsburg
University

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF AUGSBURG

Bachelor's Thesis in Bachelor Informatik
at Chair of Embedded Intelligence for Health Care and Wellbeing

Calibrated and Uncertainty-aware Multimodal Emotion Recognition

Author:	Anonymous
Professor:	Prof. Dr.-Ing. habil. Björn Schuller
Supervisor:	Lukas Stappen, M.Sc.
Submission Date:	August 3, 2021

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Nuremberg, August 3, 2021 Anonymous

Abstract

Meaningful quantification of predictive uncertainty is crucial for almost any type of machine learning task. Given an input, a machine learning model produces an output, regardless to it's confidence in terms of the prediction. Almost any application relies on trustworthy predictions and there are several methods for obtaining measurements of uncertainty alongside with the actual prediction.

This work aims to investigate uncertainty quantification for continuous emotion recognition, so the prediction of an emotional state over time, and tries to purify sources of uncertainty in this setup. We provide three main contributions. First, an extensive overview over existing state-of-the-art approaches for obtaining confidence measurements is given. Afterwards, we propose an approach for transferring common techniques to quantify predictive uncertainty for single point regression tasks to contiguous time series like predictions. Finally, we introduce subjectivity among multiple human raters as potential indication for predictive uncertainty and examine its coherence with our neural network's confidence. The experiments are carried out on the MuSe-CaR dataset [1]. It consists of videos labelled by the speaker's emotional state and contains annotations from multiple different raters for each sample. The results show that predicting measurements of confidence alongside with predicting time series is feasible but may require some conceptional changes. Furthermore, we observe that raters' subjectivity does not reflect model's uncertainty on own.

Contents

Introduction 1

Imagine any forecasting model, making a prediction. A measurement of the model's confidence about its prediction is crucial in almost any application because one has to know whether the prediction is reliable or not. Therefore, it is common to apply techniques for predicting measurements for uncertainty alongside with the actual prediction. This measurement is evaluated according to its ability to represent true uncertainty.

Instead of trying to avoid predictive uncertainty by building models which are confident in every prediction, we consider predictive uncertainty as accompanying phenomenon of predictions. Hence, we restrict ourselves on approaches to quantify it alongside with the actual prediction.

1.1 Types of Uncertainty

Before we start defining and measuring predictive uncertainty, we need to break down what circumstances can cause it. Maybe one might wonder what conditions might affect predictive uncertainty and whether a well-trained neural network should not be correct and confident about each of its predictions. Kendall et al. name two main kinds of uncertainty: aleatory (or data) uncertainty and epistemic (or model/knowledge) uncertainty [2]. Aleatory uncertainty is caused by the non-determination of an estimated problem which leads to different output for similar inputs. As described in [3], there often is no ground-truth for a given task. Hence, human raters need to label data so that the model can learn them. But human labelling rarely leads to an objective truth but mostly rather to subjective opinions. Considering multiple human-made labels implies a (high) degree of uncertainty. And if the data itself contains uncertainty, the model adopts it. Aleatory uncertainty is interwoven with noise. One speaks of noise (in terms of a pair of input features and a certain label) if the features do not transport clear information on the target. So, features (or samples), that allow multiple labels, are noisy by definition. Epistemic uncertainty, on the other hand, describes the fact that a model produces output for a given input sample whether it had been trained on comparable samples or not (e.g., an image of a ship fed to a cat-vs.-dog classifier).

1.2 Research Questions

In this work, we want to measure predictive uncertainty in contiguous[1] emotion recognition. Furthermore, we investigate the impact of subjectivity among multiple annotators on predictive uncertainty. Therefore, we define true uncertainty twice: predictive performance (as representation of trustworthiness) and disagreement among raters (as indication of subjectivity). Afterwards, we compare predicted uncertainty with both of these definitions for true uncertainty. In summary, these are our research questions:

1. Can trustworthy uncertainty measurements be obtained for emotion recognition? If so, how?

2. Does the raters' subjectivity in emotion recognition represented quantified uncertainty?

To do that, we also need to introduce approaches that let us transfer common techniques to quantify predictive uncertainty to uncertainty quantification for time series like predictions. This also could be seen as a point of research because there has not been done much research for this yet.

[1] Instead of a single value, we predict time series (so the emotional state over time).

Background 2

This chapter introduces to emotion recognition with deep learning and brings in the relevant concepts. We start with a brief explanation on machine learning and deep learning, how it works, and how it can be applied for problem-solving (i.e., emotion recognition).

2.1 Deep Learning

This section starts by giving an intuition on what models are on a high-level and explains the most important terminology. Afterwards, we describe neural networks more formally and present more advanced Deep Learning architectures. This section only gives a brief intuition on this concepts. For more detailed explanations, please refer to [4].

2.1.1 Formalization

The term Artificial Intelligence describes methods for automating intelligent behaviour or decision-making. However, this does not need to be real intelligence as it can also mean simulated or pretended intelligence. Machine Learning is a group of algorithms that are used to build Artificial Intelligence systems. These algorithms use large amounts of data to obtain knowledge from it and are able to apply this knowledge to new (unseen) data points. Neural Networks (NNs) are a special subset of Machine Learning algorithms that are inspired by the structure of the organic (e.g., human) brain. There, nodes and weights between nodes are arranged in layers and a node is connected to each node in the following layer by one weight per connection.

An input vector x is processed through a NN $f(x;\theta)$, by multiplying (matrix multiplication) x by the weight vector θ_1 of the first layer (the input layer). Afterwards, the result vector of this operation is multiplied with the weight vector of the next layer θ_2. This repeats until each layer of the NN is passed and the final layer (the output layer) is reached. The final result vector is the output $\hat{y}$ of the model. Layers between input and output layer are known as hidden layers. Mostly, a so called activation function is applied to each neuron, before the values are passed to the next layer.

This causes regularization, for example by avoiding arbitrarily large values. Deep Learning (DL) models are NNs with multiple hidden layers.

The process above, describes a forward run[1] of a simple feed-forward NN. But yet, we did not mention what the actual values of the weights θ are. At the beginning, the weights are just arbitrarily initialized numbers and the generated output $\hat{y}$ is not informative. For optimizing θ on a data set $(X; y)$ of pairs of features x_i and labels y_i, we need to define a loss function $L(y_i; \hat{y_i})$ that becomes small for predictions (so the model outputs $\hat{y_i} = f(x_i, \theta)$) which fulfill a certain criterion, and becomes large for predictions that do not fulfil it. Often, the absolute distance between y and $\hat{y}$ is used as such criterion[2]:

$$L(y_i, \hat{y_i}) = |y_i - \hat{y_i}| \tag{2.1}$$

Now, we can compute the gradient of the loss function with respect to the model weights $\frac{\delta L}{\delta \theta}$. We have to do this for each $\theta_{i,j}$ (j-th weight between layer $i-1$ and i) and as the gradients of θ_{i-1}s depend on the θ_is' gradients, this way of computing gradients, from the output layer back to the input layer, is called Backpropagation. Finally, we update the model parameters (i-th iteration) with:

$$\theta^{i+1} = \theta^i - \alpha * \frac{\delta L}{\delta \theta^i} \tag{2.2}$$

Here α denotes the learning rate which is a hyperparameter to control the impact of a single update procedure. This concept is known as Gradient Descent because we let θ descend inside the optimization landscape of L. This causes that (hopefully) the loss becomes smaller for each iteration. For the loss function, as defined above, this would cause that the prediction gets more similar to the label.

There are two major kinds of prediction tasks: classification and regression. Classification means predicting one (or more) class assignment from multiple possibilities (e.g., the kind of an animal on a photo), as while regression refers to the prediction of a numeric value (e.g., the price of a house).

2.1.2 Advanced Architectures

In this section, we introduce two main modifications from classical feed-forward NNs, namely Recurrent Neural Networks (RNNs) and Attention mechanism. Convolutional Neural Networks [5] are also very common but they do not appear in our experiments. Therefore, we don't pay more attention to them. The idea behind RNNs is to enable NNs to process temporal data. The basic intuition is that the features of each point in time are fed into the RNN layer step-by-step together with the

[1] We passed on input forward, transforming it to output.

[2] Because we want the model's prediction to match the label.

layer's hidden state from the previous time step. This enables the model to remember information from previous points in time. Problems encountered by RNNs are known as vanishing and exploding gradients, especially for longer sequences. Long Short-Term Memory (LSTM) tries to overcome this issue by incorporating additional weights to control the flow of the gradient over time [6]. Attention layers are able to learn what parts of the input features (e.g., words from a sentence) they should pay attention to [7]. This means, an attention layer is forced to learn the importance that should be given to a part of the input. For instance, this approach is widely used for text-to-text tasks (like neural machine translation). When predicting the next word w of the output sentence, the attention mechanism is used to determine the amount of importance each word of the original sentence has for the prediction of w.

2.1.3 Feature sets

For DL, we want the features, fed into the model, to be as expressive as possible. That is why it is common, especially when working with complex data like text or audio, to generate feature embeddings from the raw data and afterwards, pass these embeddings into the actual model to make a prediction. In our experiments, we use Bidirectional Encoder Representations from Transformers (BERT) [8] to obtain feature embeddings from original text features and VGGish [9] for raw audio features. Both, BERT and VGGish, are neural networks themselves which have been trained on large amounts of data[3] to explicitly generate meaningful feature representations from raw input features. Such architectures are referred to as encoders, as they encode the actual features.

2.2 Emotion Recognition

We employ Russel's Circumplex Model of Affect [10] to define an emotional state. In this, valence and arousal are dimensions that span a space where emotional states can be represented as point into this space (fig. 2.1). While the sentiment of the emotion is expressed by valence, arousal corresponds to the excitement in this model.

[3] More than 3 trillion words for BERT and over 2 million soundtracks of length 10 sec. each for VGGish

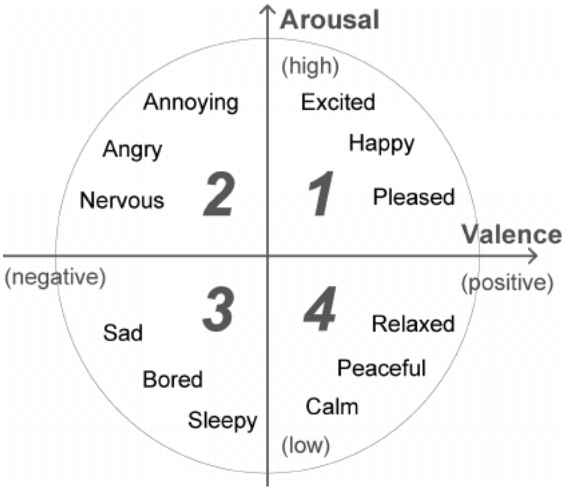

Fig. 2.1.: The Circumplex Model of Affect by Russel is a two dimensional space that represents emotions as points. In this model, an emotional state is described by its amounts of valence and arousal. Image taken from [11].

Although this provides a theoretical frame, its accuracy for practical applications is controversial (for us) as emotion recognition is highly subjective and heavily depends on interpretation. In this work, emotion recognition refers to either predicting valence or arousal.

3 Literature Review: Obtaining Reliable Uncertainty Measurements

In this chapter, we provide an extensive overview over existing state-of-the-art methods to measure and handle predictive uncertainty in DL models. Therefore, we start with approaches for measuring the confidence of a NN for a prediction. A model is called (well-) calibrated if its predictions of uncertainty correspond to true uncertainty. What true uncertainty is depends on its definition and is discussed in detail in section 3.2.

3.1 Uncertainty Quantification

There are a lot of methods for measuring predictive uncertainty of NNs. This section goes over the most common and promising approaches. Fundamentally, the core intuition behind most of them is to make the model return a measurement of uncertainty alongside with the actual prediction. For instance, when predicting affiliations of a sample to classes, one could use softmax function (eq. 3.1) to obtain probability estimations for each class from the output generated by the network.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i/T}}{\sum_{j=1}^{K} e^{z_j/T}} \quad \text{for } i = 1, \dots, K \tag{3.1}$$

T denotes the (scaling) temperature[1], $\sigma(z)_i$ the probabilistic output for the i-th class and $z = (z_1, \dots, z_K) \in \mathbb{R}^K$ the predicted output vector for K possible classes.

Further, we can differentiate between **implicitly estimating** the confidence from the model's actual output (as done with softmax for instance) and **explicitly forcing** the model to output its confidence, e.g., with an additional output node [12, 13] or by predicting distributions instead of the actual target (sec. 3.1.2).

[1] We'll refer to that later.

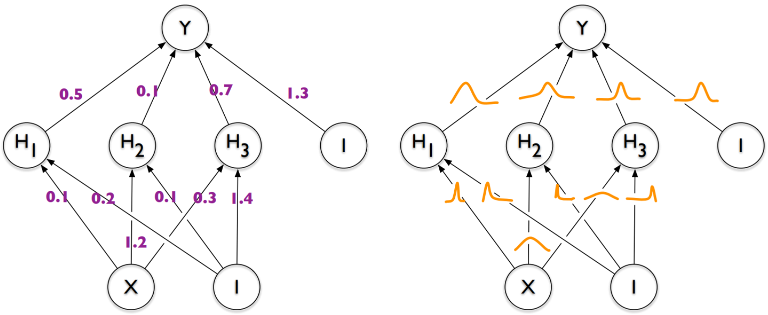

Fig. 3.1.: Learning probability distributions, instead of common (numeric) weights, enables non-deterministic computation graphs. Image taken from [15].

3.1.1 Implicit Uncertainty Measurement and Bayesian Modeling

Bayesian approaches are characterized by the non-deterministic of their computation graph. The simplest idea, for quantifying predictive uncertainty, is to calculate multiple forecasts (with non-deterministic outcome or variational inference, respectively) and use their mean (eq. 3.2) as final prediction and the variance (eq. 3.3) among them as measurement for uncertainty. High variance suggests high uncertainty.

$$\mu(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.2}$$

$$\sigma^2(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu(x))^2 \tag{3.3}$$

These methods have the assumption in common that for predictions with high confidence, the non-deterministic (but similarly optimized and working) predictions lead to similar (low variance) outcomes. And vice-versa, the higher the uncertainty, the higher the disagreement among predictions.

A popular approach for quantifying a neural network's predictive uncertainty is Monte Carlo Dropout (MCD) [14]. For that, dropout (randomly switching off/discarding nodes) is used at prediction time as well (instead of just at training time). Blundell et. al. propose an efficient algorithm for learning probability distribution parameters as NN weights [15], as visualized in figure 3.1. Another very simple but computational expensive approach is Ensemble Averaging (or referred to as Deep Ensemble sometimes). For this, several NNs (with identical architecture) are trained from varying starting points (seeds). There are lots of modifications and contributions on this method that, for instance, improve optimization with adversarial training samples [16] or use differing hyperparameter settings for ensemble members [17].

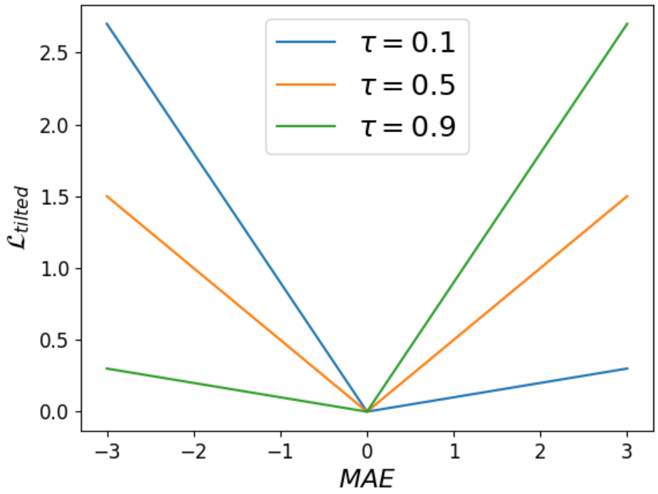

Fig. 3.2.: Tilted Loss function for the 0.1, 0.5 and 0.9 quantiles.

Quantile Regression [18] estimates quantiles $\tau_i \in [0, 1]$ with one output node i per quantile. The loss is defined by equation 3.4.

$$\mathcal{L}_{tilted_i} = max(\tau_i * e_i, (\tau_i - 1) * e_i) \tag{3.4}$$

Here, e denotes the Mean Absolute Error (MAE) (eq. 3.5).

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{3.5}$$

The Tilted Loss function is also know as Pinball Loss because it tips the error for a given by the affiliation to this quantile, as shown in figure 3.2. Measurements for uncertainty could be obtained by calculating the difference between the forecasts from the upper and the lower quantile. The closer together they are, the higher is the model's confidence.

The idea of optimizing multiple nodes on the same problem but with different loss functions gets generalized in [19]. There, Achrack et al. propose to predict the same problem with multiple losses in parallel with a different prediction head per loss function. Further, there exists an approach that makes use of the depth of modern NNs and uses the preliminary predictions of hidden layers to calculate a measurement of uncertainty [20]. Wen et al. propose a method for training multiple ensemble members in parallel (on only one device) by composing weight matrices from a shared weight matrix and a rank-one matrix per ensemble member [21].

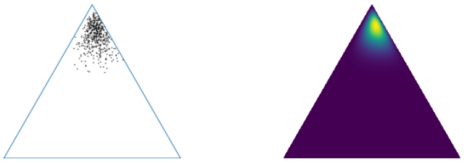

Fig. 3.3.: Distribution over distributions: the edges represent classes and the dots in the left simplex visualize predictions about class assignments (e.g., by Monte Carlo Dropout). The right simplex shows the explicitly outputted distribution over distributions from a Prior Network which behaves like the implicitly originated one on the left. Image taken from [22].

3.1.2 Digression: Higher-order Distributional Parameter Estimation

A problem of Bayesian approaches is that they often require significantly more memory or computational expensiveness. So, in contrast to Bayesian modeling, where uncertainty is measured implicitly from the model's output, it is a common approach to force a model to explicitly generate measurements of confidence.

These methods do not appear in our experiments and the knowledge of this section is not required to follow the experiments and evaluations (chapter 6). But as this work claims to give an extensive overview on state-of-the-art uncertainty measurement, we want to describe them. After the following section the reader has a solid intuition and knowledge on the latest works. One, who's only interested in uncertainty for emotion recognition and the impact of subjectivity among several annotations on it, might want to skip this section.

Malinin et al. [22] propose Prior Networks. Those are a method for obtaining NNs that generate predictions for classification, which behave like implicit distributions from Bayesian modeling approaches, with only one feed-forward run. Bayesian approaches output multiple predictions. Each of them is an estimated distribution of class assignments. Hence, what results from that is a distribution over distributions, as visualized in figure 3.3. Instead of obtaining an implicit conditional distribution (over distributions) resulting from several predictions, a Prior Network explicitly outputs such a distribution (over distributions). Further, the authors desire a specific behavior of these distributions. They differentiate between epistemic (knowledge) uncertainty and aleatory (data) uncertainty. They refer to aleatory uncertainty if a sample contains a high degree of class overlap (noise) and to epistemic uncertainty if the sample is out-of-distribution (OOD), so unknown. This differentiation has impact on the distributions. Samples with high data uncertainty show high confidence for uncertain distributions (fig. 3.4 mid.), as while predicted distributions for OOD samples range over the whole target space, so uncertainty on any value in the

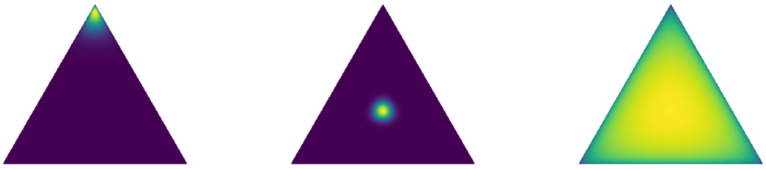

Fig. 3.4.: Desired behaviors of a distribution over distributions: high confidence (left); high confidence that uncertain, because of aleatory uncertainty (middle); high uncertainty, because of epistemic uncertainty (right). Image taken from [22].

target space (fig. 3.4 right).[2] The authors propose to implement Prior Networks by predicting a Dirichlet distribution, which is a distribution over a simplex. Here, a simplex is the space of class assignment probability distributions (like the triangles in figures 3.3 and 3.4). Such a NN is referred to as Dirichlet Prior Network (DPN) and it outputs the density parameters of the Dirichlet distribution. A prediction of a common NN with softmax activation (eq. 3.1) as output can be seen as prediction of the expected categorical distribution from a Dirichlet distribution. The paper claims that there are multiple possible ways to train a DPN. It is proposed to do it in a multi-fashion way. The authors minimize the Kullback–Leibler (KL) divergence[3] between the models predicted distribution $\hat{y}$ and a sharp (certain) Dirichlet distribution focused on the appropriate class for in-distribution data alongside with the KL divergence between $\hat{y}$ and a flat (uncertain) Dirichlet distribution for OOD data. Later, Malinin et al. propose to distill an ensemble of models into a Prior NN by minimizing the KL divergence between the distribution over predicted categorical distributions by the ensemble and the DPN's output [23]. By modifying the distribution that is parameterized by the model output, Malinin et al. enable Prior Networks for regression tasks [24]. For that, they replace the Dirichlet distribution by a Normal-Wishart distribution. As for classification, an ensemble would predict sets of parameters of normal distributions (a distribution over distributions) and similar to the Dirichlet distribution for classification, the Normal-Wishart distribution acts as a higher-order distribution over the parameters of multivariate normal distributions. This enables to differentiation between types of uncertainty (fig. 3.5), as it can be done for classification (fig. 3.4).

Sensoy et al. propose Evidential Deep Learning [25] which is similar to Prior Networks. According to the authors, estimating class assignments with softmax (eq. 3.1) equals a maximum likelihood estimation and therefore, it is not capable of inferring the variance of the predicted distribution. Further, as the probability output depends on a arbitrary parameter T, it is unreliable. Summarized, softmax provides a quantification for probabilities for class affiliations but no quantification of the uncertainty belonging to this prediction. Hence, the authors treat the predictions

[2] This is the behavior the authors expect from well-calibrated Bayesian modelling approaches. They await the same behavior from the distribution predicted by the Prior Network.

[3] The KL divergence is a similarity metric for probability distributions.

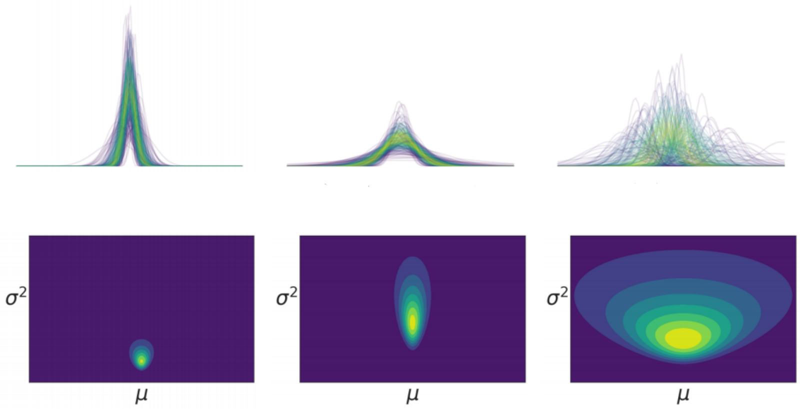

Fig. 3.5.: Desired behaviors of a Normal-Wishart distribution (bottom) over distributions (top) for Regression: high confidence (left); aleatory uncertainty, so confident about the expectation but uncertain about its correctness, same mean but high variances respectively (middle); high uncertainty because of epistemic uncertainty (right). Image taken from [24].

of a model as a subjective opinion and see the output of the model as collected evidence (that has lead to this opinion) to parameterize a Dirichlet distribution. Furthermore, the optimization does not rely on OOD samples (as it is the case for Prior NNs). Instead, Sensoy et al. define their losses according to the obtained Dirichlet distribution and its feasibility to represent the sample's label. Amini et al. apply a similar method on regression tasks [26].

Charpentier et al. name two main issues with Prior Neural Networks and Evidential Deep Learning. They mention that both need to specify an arbitrary (and fixed) target distribution. Besides that, Prior NNs require OOD training samples.[4] That is why, the authors propose Posterior Networks to overcome these issues (for classification) [27]. Given an input, an encoder generates low-dimensional feature representations. Further, one Normalizing Flow[5] component per class computes learned and flexible density functions. During inference, these density functions are evaluated at the position of the latent feature representation. Finally, the obtained densities are used to parameterize a Dirichlet distribution.

3.2 Confidence Calibration

Uncertainty calibration can be imagined as a fine adjustment of raw uncertainty measurements. As mentioned above, a model is called well-calibrated if its predicted uncertainty $\hat{U}$ is equal to true uncertainty U. If $\hat{U}$ does not match U innately, a

[4] For most applications the availability of OOD but still realistic samples in not given.

[5] A Normalizing Flow, is a sequence of transformations that transforms a simple probability distribution (e.g., Gaussian) into a complex one. Such transformations can be implemented using neural networks [28, 29].

calibrator can be learned to make $\hat{U}$ well-calibrated, so to (re-) calibrate it. For instance, when considering softmax (eq. 3.1) for a classification problem, we could use Temperature scaling. This means optimizing the temperature T on validation data so that the output is well-calibrated [30]. Hence, T would act as the calibrator.

But before diving deeper into approaches on how to calibrate non well-calibrated uncertainty measurements to become well-calibrated ones, we need to introduce ways for defining and measuring calibration (or well-calibration, respectively), in the next section.

3.2.1 Measuring Calibration

Two common, basic metrics for measuring the quality of a model's probabilistic prediction are the Negative log-likelihood (NLL) [31] and the Brier score [32], as given in equation 3.6 and 3.7, respectively.

$$NLL = -\sum_{t=1}^{T} log([H(x_t)](y_t)) \tag{3.6}$$

$$BS = \frac{1}{T}\sum_{t=1}^{T}(H(x_t) - y_t)^2 \tag{3.7}$$

H denotes a model, $H(x_t)$ its probabilistic output, y_t the true probability and T is the number of samples. However, these metrics measure the quality (or correctness) of the prediction according to its confidence, but not the quality of the actual confidence measurement itself.

Guo et al. [30] define a model for classification tasks as perfectly calibrated if the confidence $\hat{P}$ of a predicted class assignment $\hat{Y}$ equals the true probability p for the class Y being predicted correctly ($\hat{Y} = Y$). Or more formally:

$$\mathbb{P}(\hat{Y} = Y | \hat{P} = p) = p \quad \forall p \in [0, 1] \tag{3.8}$$

For example, $\mathbb{P}(\hat{Y} = Y | \hat{P} = 0.8) = 0.8$ implies that predictions with 80% confidence are correct for 80% of the times. A common metric for measuring the quality of calibration is the Expected Calibration Error (ECE) (eq. 3.9) which decomposes the confidences in bins and compares the accuracy and the confidences per bin.

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)| \tag{3.9}$$

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i) \tag{3.10}$$

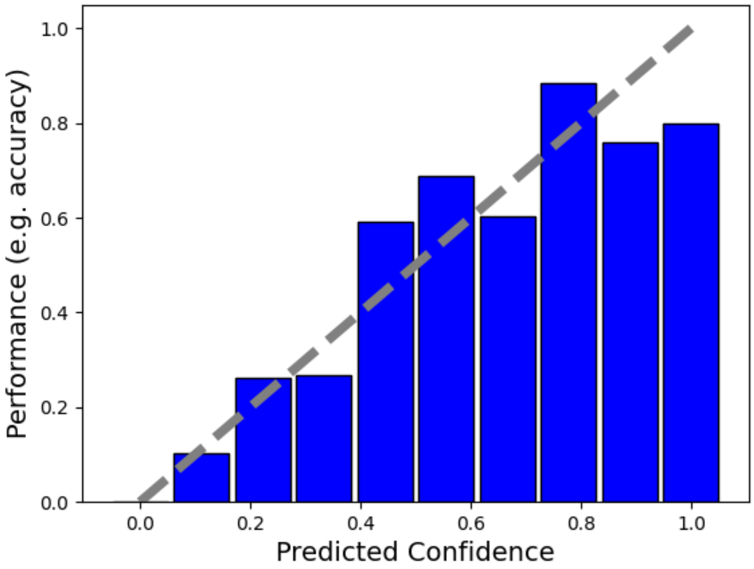

Fig. 3.6.: Relativity Diagrams: if the height of the bins match the gray diagonal, the model is perfectly calibrated.

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{3.11}$$

For this, B_m refers to the m-th bin. Further, one can visualize calibration with Reliability Diagrams by plotting the model's accuracy as a function of its confidence, as seen in figure 3.6. Laves et al. propose a modification to ECE. They define true uncertainty as the prediction error and the predicted uncertainty as entropy of predictions [33]. Further, Nixon et al. show several deviations that extent ECE, which was originally designed for binary classification, to multi-class and eradicate issues coming along with binning [34].[6]

For regression tasks, there is no such undisputed definition for well-calibration as given for classification with equation 3.8. However, the most common one comes from Kuleshov et al. In their paper, they propose a interval-based method for measuring and re-calibrating uncertainty estimates for regression [35]. Well-calibration is achieved if a true label y falls into the $p\%$ interval for approximately $p\%$ of the times.[7]

Levi et al. define and measure uncertainty for regression in a different manner [36]. They dispense on measuring percentage information of the confidence. Their definition for confidence is more in line with that from Guo et al. for classification

[6] They align the width of bins according to the number of included samples, instead of fixed widths, etc.

[7] Compare to figure 3.7.

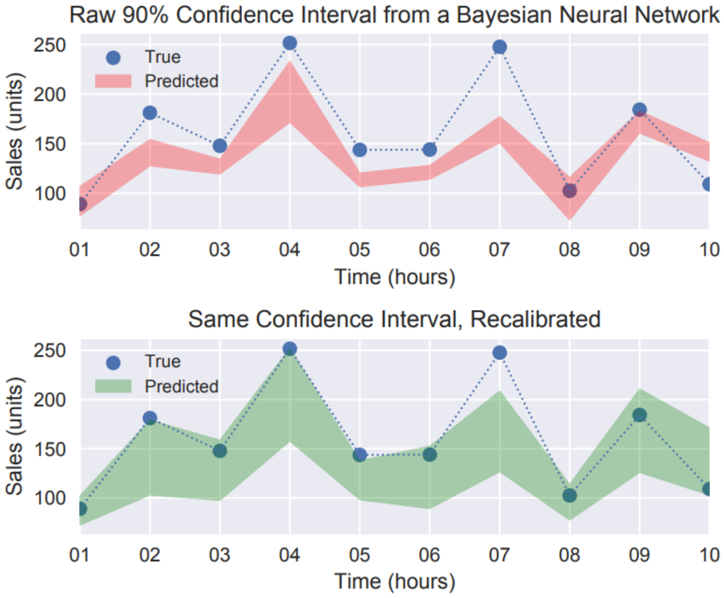

Fig. 3.7.: The upper graph shows the 90% confidence interval (red), generated by a Bayesian neural network. But it fails to hold the conditions for well-calibration as less then 90% of actual predictions fall into the interval. The lower plot is obtained by re-calibrating the upper one so that 9 out of 10 predictions fall into the (green) interval. Image taken from [35].

(eq. 3.8). The quality of calibration is measured by the accordance of predicted uncertainty and true uncertainty. To do so, they replace accuracy with prediction error and the percentage confidence with variance (eq. 3.12).

$$\forall \sigma : \mathbb{E}_{x,y}[(\mu(x) - y)^2 | \sigma(x)^2 = \sigma^2] = \sigma^2 \tag{3.12}$$

Here, $\mu(x)$ and $\sigma(x)^2$ denote the model's actual prediction (e.g., the mean of several deep ensemble members) and its generated variance, respectively. Intuitively, this means that a model is well-calibrated, according to Levi et al., if the prediction error equals the predicted uncertainty. The Expected Normalized Calibration Error (ENCE) is introduced for measuring the summarized calibration error by binning[8] the predicted uncertainties (eq. 3.13).

$$ENCE = \frac{1}{N} \sum_{j=1}^{N} \frac{|RMV(j) - RMSE(j)|}{RMV(j)} \tag{3.13}$$

$RMV(j)$ denotes the root of the mean variance (eq. 3.14), so the uncertainty of bin B_j, and $RMSE(j)$ the root mean squared error (eq. 3.15) of it.

$$RMV(j) = \sqrt{\frac{1}{|B_j|} \sum_{t \in B_j} \sigma_t^2} \tag{3.14}$$

[8] Binning means pooling of similar samples in bins.

$$RMSE(j) = \sqrt{\frac{1}{|B_j|} \sum_{t \in B_j} (y_t - \hat{y}_t)^2} \tag{3.15}$$

For each bin, the absolute difference is divided by $RMV(j)$, because the authors expect (and overlook) higher calibration errors in areas of larger uncertainty.

Additionally, the authors propound to use the c_v (eq. 3.16) as second metric to support evaluation of ENCE scores.

$$c_v = \frac{\sqrt{\frac{\sum_{t=1}^{T}(\sigma_t - \mu_\sigma)}{T-1}}}{\mu_\sigma} \tag{3.16}$$

μ_σ denotes the mean over the variances σ_t:

$$\mu_\sigma = \frac{1}{T} \sum_{t=1}^{T} \sigma_t \tag{3.17}$$

Imagine a model that is not capable of estimating true uncertainty but outputs a fixed value which coincidentally matches the mean of prediction errors (so the true uncertainty). ENCE would be equal to 0. Hence, c_v can be used to confirm that the model actually distinguishes between lower and higher uncertainty (c_v not small) and should be used alongside with ENCE.

3.2.2 (Re-) calibration

In this section, we present methods for transforming non well-calibrated uncertainty approaches into well-calibrated ones. Intuitively, there are two basic kinds of procedures: modifying the process of uncertainty quantification, so that the quantified uncertainty is well-calibrated innately, or learning an auxiliary model (a calibrator) to make raw, uncalibrated confidence estimates well-calibrated. Usually, one wants to use (hold-out) validation data to fit the calibrator.

Guo et al. name several approaches for re-calibration [30]. For Histogram Binning, the uncalibrated scores are grouped into bins and a (fixed) calibration score (which is obtained by evaluating validation data, for instance) is assigned to each bin. Platt scaling means the parametric transformation of uncalibrated scores [37]. For example, Temperature scaling, as described at the beginning of section 3.2, is an extension of Platt scaling for classification with NNs. Other parametric calibration methods learn parameters so that an underlying function, given an uncalibrated Probability Density Function (PDF), matches the calibrated PDF (for example logistic calibration [38] and beta calibration [39, 40]). Levi et al. [36] learn a calibrator by minimizing the NLL of the output PDF (through optimizing the calibrator's

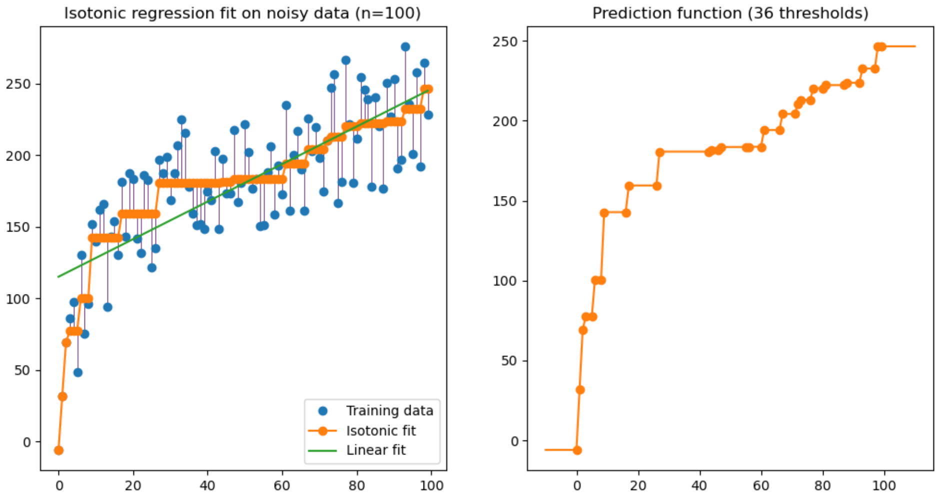

Fig. 3.8.: Isotonic Regression: this non-parametric approach optimizes its output according to equation 3.18 (orange dots). Finally, the output is interpolated (orange line) and can be used for inference. Image taken from [42].

parameters). They implement the calibrator, in its simplest form, as a single scalar value which is multiplied with uncalibrated uncertainty. Kumar et al. combine Platt scaling based methods and Histogram Binning by fitting a parametric function and binning the function values afterwards [41].

Isotonic Regression (IR) is the (probably) most common non-parametric approach. It fits a non-decreasing function to 1-dimensional data y_i (here, the uncalibrated confidences) by minimizing equation 3.18.

$$\underset{\hat{y}}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \text{ with limitation } \hat{y}_i \leq \hat{y}_j \text{ whenever } y_i \leq y_j \tag{3.18}$$

So, this calibrator outputs the calibrated scores of its training data (the validation data actually). Those outputs $\hat{y}_i$ are interpolated so that the complete feature space is covered and can be queried during inference [42] (visualized in figure 3.8).

Thiagarajan et al. [43] propose to obtain calibrated confidence estimations during training by Uncertainty Matching. This technique forces the quantified uncertainty estimation to match the actual uncertainty (i.e., to be calibrated). They train two models separately. One of them (the main model) outputs a mean $\hat{y}$ (the prediction for the actual target) and quantifies a confidence interval[9] σ. The second (auxiliary) model outputs lower and upper interval borders δ^l and δ^u. The loss

[9] This could be realized by any uncertainty measurement technique. The authors use heteroscedastic NNs (directly predicting the variance as measurement for uncertainty with an additional output node) and conditional quantile estimators (quantile regression).

function for the first model is defined so that the quantified uncertainty interval matches the prediction of the second model. On the other hand, the auxiliary model is optimized to output well-calibrated uncertainty intervals that represent underlying uncertainties (the prediction error of the main model). The authors claim that this bi-level optimization generates well-calibrated confidence intervals and improves the model's predictive performance as well.

In [44] Krishnan et al. introduce a new training criterion for deep learning models that provides well-calibrated uncertainty estimates alongside improved prediction performance. They add a term to the loss function that becomes large if a certain prediction is false or an uncertain one turns out to be correct.

Approach

4

As described in section 3.2, true uncertainty commonly is defined as the prediction error for the actual target in literature. We also keep this procedure but we extend our experiments with an additional definition for true uncertainty. Since the dataset contains multiple annotations for each sample, we use the subjectivity among those as representation for true uncertainty. We expect the model to output larger uncertainty if the sample was rated differently by the annotators. Further, we consider as sample as noisy and difficult to predict if disagreement among raters is large.

At the beginning of section 3.1.1, we mention that uncertainty is usually measured as the variance of several suggested predictions. Before we start, we need to replace variances by correlations. We predict steps in time of a contiguous emotional state. The relative movement between two (or more) opinions on it is more important than the actual value for the emotional state because time steps are not rated isolated. Imagine the annotation of a contiguous progression. Each point in time t is rated based on the evaluation of the signal at t and the annotation of the latest time step $t-1$, as visualized in figure 4.1. This gets even more clear if we anticipate the annotation procedure used for the data set of our experiments, as described in section 5.1. The human annotators used joysticks (up or down) to create the contiguous labelling over time. So when determining their annotation, they actually have been forced to evaluate each time step relatively to the last one.

One way of measuring correlation between two (time) signals x and y is the Pearson Correlation Coefficient (PCC) (eq. 4.1).

$$Corr(x,y) = \frac{Cov(x,y)}{\sqrt{\sigma^2(x)}\sqrt{\sigma^2(y)}} = \frac{\sigma_{x,y}}{\sigma_y \sigma_y} \tag{4.1}$$

σ denotes the standard deviation (square root of the variance) and the Covariance is given with equation 4.2.

$$Cov(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu(x))(y_i - \mu(y)) \tag{4.2}$$

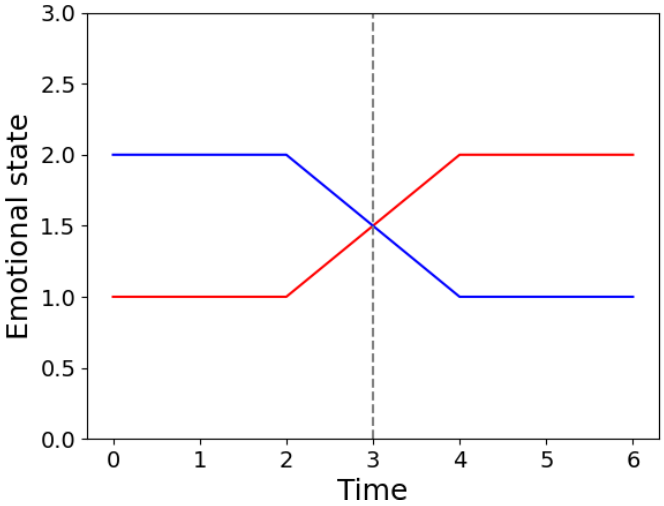

Fig. 4.1.: At $x = 3$ both lines have the same value. But when it comes to measuring the disagreement of them, the correlation at this point is much more expressive, especially because the shown values at each step in time come from one contiguous measurement. Each point in time is not a standalone prediction. Although variance ($\sigma = {}^1\!/\!_2((1.5 - 1.5)^2 + (1.5 - 1.5)^2)$) is 0 at $x = 3$, the both lines show completely different opinions on the emotional state at $x = 3$.

Additionally, we specify the Concordance Correlation Coefficient (CCC) [45] in equation 4.3.

$$CCC(x, y) = \frac{2\ Cov(x, y)}{\sigma^2(x) + \sigma^2(y) + (\mu(x) - \mu(y))^2} \tag{4.3}$$

Both, PCC and CCC are limited to $[-1, 1]$ and while -1 indicates total negative correlation, $+1$ indicates perfect correlation.

For clarity: we replace the procedure of uncertainty quantification. But for the actual prediction target, we stick to the conventional idea of using the mean (eq. 3.2) of multiple predictions. This means, when implementing a technique for uncertainty measurement that yields several forecasts (MCD for instance), we take their mean (per time step) as final prediction.[1] This happens independently of any uncertainty measurement as concrete calculations for uncertainty quantification (presented in sections 4.1 and 4.2) are applied after the final prediction (the mean) has been computed.

4.1 Local Uncertainty Measurement

We aim to measure short (or local) differences in the signals as measurements of confidence. Here, local means measurement per time step, in contrast to measure-

[1] Look at the (dark) blue line in figure 6.2 as an example. There exist multiple predictions (those are not visible in this plot) and the blue line is the mean of them at each step.

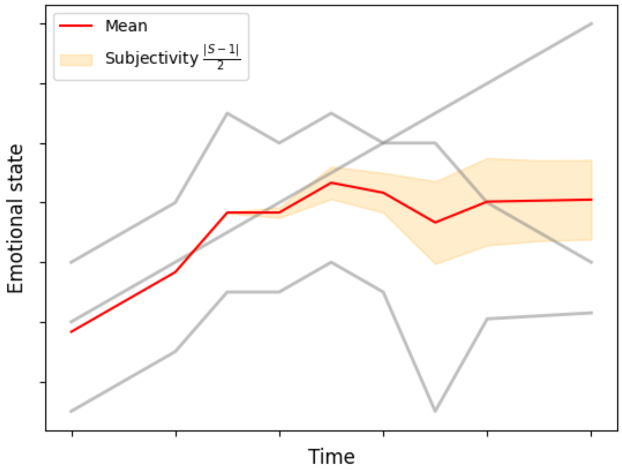

Fig. 4.2.: The gray lines represent different annotations. The orange area is $\frac{|S-1|}{2}$ with the mean of the rolling correlations S, as defined in equation 4.4.

ment for multiple time steps together (e.g., a complete sample) which is introduced in section 4.2.

4.1.1 True Uncertainty

We employ two definitions for true uncertainty: the prediction performance representing the quality of the model's prediction for an emotional state and the subjectivity among annotations. Intuitively, we define the subjectivity among annotations as the mean of the rolling correlations between all available pairs of annotations per time step. Let $x_1, \ldots, x_m$ denote the annotations from m different raters and each x_i consists of n steps of time. So, $x_{i,1} \ldots x_{i,n}$ denotes the prediction of the i-th rater and $x_{i,j}$ references the prediction of the i-th rater for j-th point in time, respectively. Subjectivity among raters of a sample at time step j is defined as:

$$S_j = \frac{1}{(m-1)+\cdots+1} \sum_{i=1}^{m} \sum_{k=i+1}^{m} Corr(x_{i,j-t} \ldots x_{i,j}, x_{k,j-t} \ldots x_{k,j}) \text{ with } t = 2 \tag{4.4}$$

Equation 4.4 expresses that the subjectivity S_j at time step j is defined as the mean of correlations between annotations within the 3 (as $3 = t + 1$ and $t = 2$) latest time steps ($j - 2$, $j - 1$ and j). This approach is visualized in figure 4.2.

Similar to that, we define the predictive performance as rolling correlation (compare figure 4.3) between label y prediction $\hat{y}$:

$$P_j = Corr(y_{j-t} \ldots y_j, \hat{y}_{j-t} \ldots \hat{y}_j) \text{ with } t = 2 \tag{4.5}$$

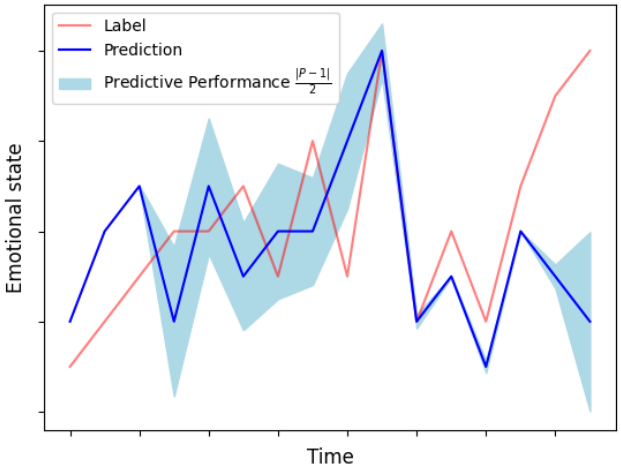

Fig. 4.3.: The red line shows the label of a sample and the blue one the prediction of a model for it. The smaller the rolling correlation (eq. 4.5) between label and prediction, as larger the light-blue uncertainty ($\frac{|P-1|}{2}$).

Finally, we can complete the definitions for local uncertainty of a complete sample:

$$S = S_3, S_3, S_3, S_4, \ldots, S_n \qquad (4.6) \qquad\qquad P = P_3, P_3, P_3, P_4, \ldots, P_n \qquad (4.7)$$

S as well P are based on the PCC and therefore, they are also limited to $[-1, 1]$. Note, **high variance has indicated high uncertainty. But now high correlation reflects low uncertainty**. So, for obtaining descriptive visualizations for S and P as uncertainty (fig. 4.2 and 4.3), we need to reverse the magnitude by subtracting 1, taking the absolute value and (optionally) dividing by 2.

At this point we should mention that the term subjectivity (here) behaves a bit odd. For instance, when S_j (eq. 4.4) becomes $+1$ (maximum), the annotators totally agree. But this means that *subjectivity*, as intuitive expression, is actually minimal. So note, the larger the subjectivity (calculation) is, the higher is the (expected) confidence (although the name might suggest the opposite).

4.1.2 Predicted Uncertainty

In the last section, we defined true uncertainty. We employ several approaches to predict it alongside with the emotional state.

With Monte Carlo Dropout[2] we obtain a forecasts $\hat{y}_1, \ldots, \hat{y}_a$. This leads to a similar situation as faced in the last section for m annotations. The predicted uncertainty by MCD is defined as:

$$MCD = MCD_3, MCD_3, MCD_3, MCD_4, \ldots, MCD_n \tag{4.8}$$

With:

$$MCD_j = \frac{1}{(a-1) + \cdots + 1} \sum_{i=1}^{a} \sum_{k=i+1}^{a} Corr(\hat{y}_{i,j-t} \ldots \hat{y}_{i,j}, \hat{y}_{k,j-t} \ldots \hat{y}_{k,j}) \text{ with } t = 2 \tag{4.9}$$

Further, we want to apply Quantile Regression. But the Pinball Loss (eq. 3.4), which is the loss function for Quantile Regression, is based on the MAE (eq. 3.5) and previous experiments of us have shown that this is not a suitable optimization criterion for continuous emotion recognition.[3] In summary, the reason is that if fitted with MAE, the model tends to predicting a straight line at the mean of the label[4] because often this might be a good fit in terms of MAE as it measures the difference at each point separately. But this is not desirable because the emotional state, as a whole, is not recognized. Hence, we propose tilted CCC (tCCC) (eq. 4.12). It is a loss function which provides measurements of uncertainty and still uses CCC (or any other loss function) as foundation.[5] For tCCC, we equip the model with two additional output nodes. Therefore, the prediction of each time step $\tilde{y}_j$ consists of three values $\tilde{y}_{j,1}, \tilde{y}_{j,2}, \tilde{y}_{j,3}$. tCCC is just the sum of individual loss functions for each node. The middle node[6] is optimized using CCC loss (eq. 4.11) and is used to predict the emotional state. Nodes one and three are used to quantify the model's confidence. One of them is optimized by the rolling correlation error (eq. 4.10) of the w_l latest time steps and the other by the rolling correlation error of the w_u ($w_u > w_l$) latest time steps. The middle node is used for inference of the emotional state and the rolling correlation between prediction of the first and the third node is taken as measurement of confidence (eq. 4.13).

$$RCE_w(\hat{y}) = \frac{1}{n} \sum_{j=w}^{n} (1.0 - Corr(\hat{y}_{j-w+1} \ldots \hat{y}_j, y_{j-w+1} \ldots y_j)) \tag{4.10}$$

$$CCCLoss(\hat{y}) = 1.0 - CCC(\hat{y}, y) \tag{4.11}$$

[2] Details on the implementation and configuration are given in chapter 5.

[3] Anonymous

[4] So, $\hat{y}_j \approx \mu(y), \forall j = 1 \ldots n$

[5] Although it actually doesn't tilt anything, like the pinball loss, we call it tilted CCC as it is derived from tilted loss.

[6] It could be the first or the third as well. The order does not matter.

With it, tCCC can be written formally:

$$tCCC_{l,u}(\tilde{y}) = \frac{RCE_{w_l}(\tilde{y}_{1,1} \dots \tilde{y}_{n,1}) + CCCLoss(\tilde{y}_{1,2} \dots \tilde{y}_{n,2}) + RCE_{w_u}(\tilde{y}_{1,3} \dots \tilde{y}_{n,3})}{3} \tag{4.12}$$

For inference, we can calculate the measurement of confidence like the predictive performance:

$$T_j = Corr(\tilde{y}_{j-t,1} \dots \tilde{y}_{j,1}, \tilde{y}_{j-t,3} \dots \tilde{y}_{j,3}) \text{ with } t = 2 \tag{4.13}$$

$$T = T_3, T_3, T_3, T_4, \dots, T_n \tag{4.14}$$

Classical Quantile Regression creates a mini-ensemble (with one model) by fitting multiple nodes specialized on different kinds of sources of error. While the lower quantile is forced to focus on errors with negative sign, the upper quantile amplifies MAE with positive sign (compare fig. 3.2). We adopt this feature with tCCC by the first and the third node. The first shall focus on errors over very short periods (x_l time steps) and the third one on errors over a longer portion (w_u time steps). Both nodes supply opinions during inference. The correlation of those, can be used as measurement for confidence (eq. 4.13).

4.2 Global Uncertainty Measurement

We also measure uncertainty on a global level. Hence, we define uncertainty and subjectivity in a novel way. Instead of computing them for a time step as correlation of few latest points, we measure (one value of) uncertainty for a whole sample. Furthermore, we split up samples into non-overlapping sub-samples of fixed length π. For clarity, note that the samples are not split up before or during training and inference of the NN. After obtaining the predictions, we split up labels and predictions to calculate our measurements of uncertainty on these sub-samples.[7] The underlying intention is that uncertainty might change over time and quantifying one value of confidence for a complete video seems to be imprecise. Further, we replace PCC by CCC in our calculations.

[7] This means for the following formulas that the annotations $x_1, \dots, x_m$, the label y, multiple predictions $\hat{y}_1, \dots, \hat{y}_a$ and the final prediction $\hat{y}$ (which is the mean of $\hat{y}_i$s, as explained in the introduction of this chapter) refer to sub-samples of length π.

4.2.1 True Uncertainty

Let again $x_1, \ldots, x_m$ denote labels from m different annotators with length π each. The subjectivity among annotations on a global level is defined as:

$$S = \frac{1}{(m-1) + \cdots + 1} \sum_{i=1}^{m} \sum_{k=i+1}^{m} CCC(x_i, x_k) \tag{4.15}$$

and the predictive performance is

$$P = CCC(y, \hat{y}) \tag{4.16}$$

for the merged label[8] y and the models final prediction for the emotional state $\hat{y}$.

4.2.2 Predicted Uncertainty

Similar to subjectivity, the predicted uncertainty is defined as the CCC between a forward runs $\hat{y}_1, \ldots, \hat{y}_a$ for MCD and predictions $\hat{y}_1, \ldots, \hat{y}_a$ from a models trained from varying seeds for Ensemble Averaging, respectively.

$$MCD/Ensemble = \frac{1}{(a-1) + \cdots + 1} \sum_{i=1}^{a} \sum_{k=i+1}^{a} CCC(\hat{y}_i, \hat{y}_k) \tag{4.17}$$

4.3 Metrics

We employ several metrics and visualizations to measure the quality on the quantified uncertainties. First, note again that high correlations represent low uncertainty. We adopt the ENCE (eq. 3.13) by transferring its functionality from variances to correlations. For this, we replace the prediction error $(y_t - \hat{y}_t)^2$ by a general representation of true uncertainty U, which (in our case) either is subjectivity among raters or predictive performance, in equation 3.15 and the variance in equation 3.14 by predicted uncertainty $\hat{U}$. Furthermore, we need to transform U and $\hat{U}$. Elsewise each bin would be divided by $\sqrt{1/|B_j| \sum_{t \in B_j} \hat{U}_t}$. But for classical ENCE this corresponds to equation 3.14 which becomes larger for larger uncertainties (variances).[9] To maintain this symmetry, while working with correlations, we

[8] More details on the actual label, calculated from the multiple available annotations $x_1, \ldots, x_m$, can be found in section 5.1.

[9] Remember section 3.2.1: ENCE is weighted less for large predicted uncertainties.

compute $U' = \frac{|U-1|}{2}$ and $\hat{U}' = \frac{|\hat{U}-1|}{2}$, respectively.[10] This leads us to the following definition for our General Uncertainty ENCE (gENCE):

$$gENCE = \frac{1}{N}\sum_{j=1}^{N}\frac{|\sqrt{1/|B_j|\sum_{t\in B_j}\hat{U}'_t} - \sqrt{1/|B_j|\sum_{t\in B_j}U'_t}|}{\sqrt{1/|B_j|\sum_{t\in B_j}\hat{U}'_t}} \tag{4.18}$$

As already mentioned in section 3.2.1, ENCE (so also gENCE) could deliver falsely good-looking results and should be used alongside with a volatility metric. Therefore, we calculated the variance of both, predicted and true, uncertainty.[11] In addition to that, we measure the MAE between true and predicted uncertainty and, as we are predicting contiguous points in time, we calculate the CCC between them. We apply the same set of metrics for short-term and global uncertainty measurement. Finally, to be able to confirm observations visually, we create Reliability Diagrams, as introduced in section 3.2.1. To fulfill the shift from percentage confidences to correlations, we need to switch the axis ranges to $[-1, 1]$ as explained in section A.1.

4.3.1 Benchmarking

To get a reliable picture of the quality of uncertainty predictions, we compute the same metrics for randomly generated predictions for each approach. Those are generated normally distributed, parameterized by the mean and variance of the true uncertainty (subjectivity or predictive performance) U.

$$\hat{U}_{BM} \sim \mathcal{N}(\mu(U), \sigma^2(U)) \tag{4.19}$$

Approaches, whose performance is not significantly better than that of $\hat{U}_{BM}$, can be seen as incapable.

4.3.2 Calibration

We calibrate the predictions for uncertainty with IR (sec. 3.2.2) and learn the function to map predicted uncertainty to true uncertainty on validation data. Afterwards, we use the mapping to calibrate predicted uncertainty of the test set. For us, calibration turned out to be a useful tool to measure the performance of uncertainty prediction. Because only if there is any correlation between predicted and true

[10] Example: $U = -1$ (negative correlation) → $U' = 1$ (maximum uncertainty) and $U = +1$ (positive correlation) → $U' = 0$ (maximum confidence). By this, larger uncertainties are expressed by larger values, as it is the case with variances.

[11] We chose variance instead of c_v (eq. 3.16) for better interpretability. c_v can get distorted for varying means of uncertainty quantifications (eq. 3.17) which complicated comparability between different approaches for us.

uncertainty, the calibrator has any chance to find a mapping. If the variance of the calibrated predictions goes towards zero, this indicates that the calibrator was unable to learn the mapping and for this reason always predicts the true uncertainty's mean.

Experimental Setup 5

In this chapter we introduce to the MuSe-CaR dataset, details on implementation and parameterization of the approaches, described in chapter 4.

5.1 The MuSe-CaR Dataset

The MuSe-CaR dataset [1] contains, amongst other things, about 40 hours of English vehicle review videos from YouTube. The examples are annotated continuously (one label per time step) according to their emotional dimensions, valence and arousal, by at least 5 annotators. While watching the videos (audio and visual features), the raters used joysticks to create the annotations with a frequency of 0.25 Hz (4 measurements per second). Furthermore, a gold-standard is computed, using fusion techniques, which calculate a mean while weighting annotations by their deviation from the others. Thus, outliers are not weighted too much and do not wreck the overall picture.

To reduce complexity, we restrict ourselves to one feature set per emotional dimension. Based on the results from the baseline paper [1], when predicting valence, we always use BERT features [8] and for arousal, the model is given features extracted by VGGish [9].

5.2 Experimental Settings

For our experiments, we build upon the model presented in [46]. The network (with our configuration) consists of one self-attention layer with 4 heads, followed by two bidirectional-LSTM layers with 64 hidden nodes each and closes with one linear fully-connected output layer[1] with also 64 hidden nodes. The model is optimized using CCC loss (eq. 4.11) as convergence criterion.

[1] With one output node (except for tCCC)

5.2.1 Parameterization of Approaches

Further, we employ Dropout with probability 0.5 on the second LSTM layer and the final output layer. For MCD, we calculate $a = 5$ forecasts to obtain confidence and for Ensemble Averaging, we train the model $a = 5$ times (starting from different seeds).

tCCC (eq. 4.12) is parameterized by $w_l = 3$ and $w_u = 10$. So, one node focuses on the prediction error over 3 time steps and the other over 10. We require both window lengths to be small so that they are still expressive in terms of local uncertainty measurements.

Further, the length of sub-samples π, to investigate as globally (sec. 4.2), is set to 20. 20 time steps correspond to 5 seconds (0.25 seconds per time step).

In our experiments, for gENCE (eq. 4.18), the number of bins B_j is set to $N = 5$ and the bins are equally distributed over the target space ($[-1, 1]$) of the correlation function (PCC or CCC).

5.2.2 Setup

The experiments[2] are programmed with Python Programming Language[3]. The NNs are built with PyTorch[4] and the Hugging Face[5] library. Additionally, we use scikit-learn[6] for Isotonic Regression, Matplotlib[7] for any plots as well as Pandas[8] and NumPy[9] for data handling and preprocessing. Due to the high-dimensionality of the data, the experiments were executed on a GeForce GTX Titan X (12GB).

[2] Anonymous

[3] `https://www.python.org/`

[4] `https://pytorch.org/`

[5] `https://huggingface.co/`

[6] `https://scikit-learn.org/`

[7] `https://matplotlib.org/`

[8] `https://pandas.pydata.org/`

[9] `https://numpy.org/`

6 Results and Evaluation

In this chapter, we present and investigate the results from the experiments, described in chapters 4 and 5. For this, we start with the experiments that measure the model's uncertainty per time step (i.e., locally). Furthermore, we then move from local to global uncertainty. This allows the model to mint several reliable opinions which can be used to form the predicted confidence.

The results are structured in tables by emotional dimension (valence or arousal) and definition for true uncertainty (subjectivity or predictive performance). Each table contains columns for the metrics as described in section 4.3. Further, the tables for short-term uncertainty measurement (sec. 6.1) include the CCC scores for the prediction of the actual emotional state CCC_{emo}. Later, we waive this information because it does not change or impact uncertainty evaluation. Additionally, we point out the results for the randomly generated benchmark (sec. 4.3.1), denoted by [BM], to each approach. Furthermore, *cal.* refers to calibration. As mentioned, we present results for uncertainties once uncalibrated and once calibrated with IR.

The results show that CCC is (for us) the most expressive score. We observe that most (especially calibrated) uncertainty will be around 0 and with low variance. Hence, gENCE is low which is basically fine but meaningless in such cases (compare sec. 3.2.1). However, CCC does not lie. It is not very accurate but only begins to strike if there indeed exists correlation between prediction and true confidence. Together with MAE, these metrics give a solid understanding of the performance of an approach.

6.1 Local Uncertainty Measurement

The model's confidence for each time step is measured based on its very latest predictions. The tables 6.1, 6.2, 6.3 and 6.4 show the results of short-uncertainty measurement for each possible combination of definitions for true uncertainty and actual prediction target, emotional state respectively (i.e., predictive performance with valence, predictive performance with arousal, subjectivity among raters with valence and subjectivity among raters with arousal). Overall, the results show that true confidence could not be predicted successfully. CCC is always around zero and

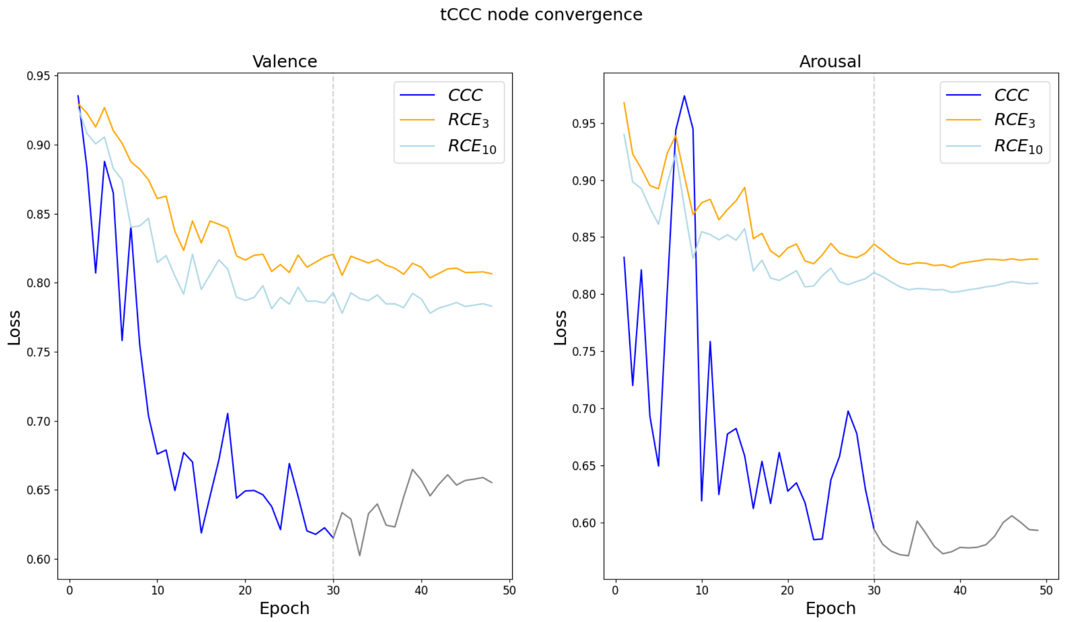

Fig. 6.1.: The graphs show the convergence of tCCC for each output node during training. For the first 30 epochs, each node contributes to the overall loss criterion. Afterwards, the blue (middle) node (which predicts the actual emotional state) becomes detached from the backpropagation-computation graph and the NN is not optimized according to the loss this node anymore. Initially, we would expect to observe convergence of the outer (orange and light blue) nodes independently of convergence of the middle node. But the opposite occurred. When the blue graph stops converging, the others also stagnate. This indicates that the model is not capable of learning short-term (3 or 10 time steps) forecasts. And if the model's short-term predictions are prone to be wrong, so meaningless, they naturally lead to non-reliable confidence estimates. For the first 30 epochs, orange and light blue probably do also converge because improvement of the predictions globally (blue node with CCC loss) implicitly implies improvement locally on average.

none of the other metrics indicates at least any success of the predictions.[1] *But why is that?* We assume, the reason, that all implicit measurements for confidence fail, does not come from the definitions of confidence or the approaches for measurement themselves. Rather it is that the whole model, even in terms of predicting the actual emotional state, does not work for very short periods (or only a few time steps, respectively). This can be observed by looking at the loss convergence for tCCC during training (fig. 6.1). As the uncertainty quantification has to rely on the model's forecasts, their malfunction explains the poor quality of predicted uncertainty.

The tCCC node convergence explains issues of this model with this data set in general and especially the poor results of quantifying uncertainty defined as the predictive performance (tables 6.1 and 6.2). But it does not necessarily clarify the

[1] gENCE has to be read carefully in such situations. This metric is useful for yet solid uncertainty estimates to measure their actual level of calibration. For predictions, that are not calibrated at all or whose quality of uncertainty prediction is very weak, gENCE is barely meaningful (compare sec. 3.2.1).

method	cal.	CCC_{emo}	gENCE	MAE	CCC	σ^2_{pred}
MCD [BM]	-	.4312 / .6063	.3021 / .3060	.8576 / .8612	.0010 / .0002	.5493 / .5569
MCD	-	.4312 / .6063	.0822 / .0950	.6857 / .6841	.0087 / -.0131	.0216 / .0220
	IR	.4312 / .6063	.1465 / .0590	.6922 / .6834	-.0079 / -.0064	.0155 / .0029
tCCC [BM]	-	.4354 / .5636	.3142 / .3174	.9781 / .9590	.0034 / -.0017	.7138 / .6862
tCCC	-	.4354 / .5636	.0800 / .1244	.8214 / .8206	-.0064 / -.0081	.0073 / .0063
	IR	.4354 / .5636	.1582 / .1536	.7871 / .7568	-.0107 / -.0154	.0178 / .0178

Tab. 6.1.: Short-term **predictive performance** measurement (devel / test) for **valence** (BERT): The variances of true confidence σ^2_{true} are $.5521/.5541$ for MCD and $.7147/.6850$ for tCCC. For predictive performance, in contrast to subjectivity among annotations, as measurement for true uncertainty, the true uncertainty (and thus its variance) depends on the actual prediction of the model. Therefore, they vary from method to method. For indicating the validation set, we refer to devel (and to test for the test set).

method	cal.	CCC_{emo}	gENCE	MAE	CCC	σ^2_{pred}
MCD [BM]	-	.4143 / .1828	.3026 / .3005	.8637 / .8650	-.0029 / -.0039	.5572 / .5567
MCD	-	.4143 / .1828	.0792 / .0796	.6887 / .6969	-.0069 / -.0015	.0220 / .0220
	IR	.4143 / .1828	.1477 / .1499	.6980 / .7142	-.0082 / -.0038	.0183 / .0179
tCCC [BM]	-	.4251 / .2070	.3178 / .3128	.9945 / .9834	-.0018 / -.0002	.7324 / .7166
tCCC	-	.4251 / .2070	.0719 / .0310	.8336 / .8113	-.0102 / -.0057	.0121 / .0131
	IR	.4251 / .2070	.0499 / .0492	.8202 / .8048	-.0161 / -.0070	.0046 / .0046

Tab. 6.2.: Short-term **predictive performance** measurement (devel / test) for **arousal** (VGGish): The variances of true confidence σ^2_{true} are $.5568/.5588$ for MCD and $.7317/.7154$ for tCCC

method	cal.	CCC_{emo}	gENCE	MAE	CCC	σ^2_{pred}
MCD [BM]	-	.4312 / .6063	.1173 / .1210	.1851 / .1878	-.0045 / -.0007	.0271 / .0278
MCD	-	.4312 / .6063	.1488 / .1602	.2173 / .2269	.0047 / .0067	.0216 / .0220
	IR	.4312 / .6063	.1994 / .1872	.3432 / .2685	.0000 / .0000	.0001 / .0017
tCCC [BM]	-	.4471 / .5945	.1144 / .1204	.1846 / .1875	-.0019 / .0048	.0271 / .0281
tCCC	-	.4471 / .5945	.2016 / .2136	.3607 / .3786	-.0012 / -.0021	.0073 / .0063
	IR	.4471 / .5945	.1919 / .2036	.3112 / .3285	-.0004 / -.0003	.0004 / .0004

Tab. 6.3.: Short-term **subjectivity** measurement (devel / test) for **valence** (BERT): The variance of true confidence σ^2_{true} is $.0272/.0279$.

method	cal.	CCC_{emo}	gENCE	MAE	CCC	σ^2_{pred}
MCD [BM]	-	.4143 / .1828	.1188 / .1079	.1873 / .1765	-.0022 / .0059	.0277 / .0248
MCD	-	.4143 / .1828	.1621 / .1395	.2319 / .2100	.0012 / -.0042	.0220 / .0220
	IR	.4143 / .1828	.1454 / .1928	.1744 / .3365	.0009 / -.0001	.0075 / .0001
tCCC [BM]	-	.4089 / .1595	.1211 / .1121	.1871 / .1766	.0004 / .0042	.0278 / .0249
tCCC	-	.4089 / .1595	.2121 / .1906	.3764 / .3444	.0005 / -.0008	.0121 / .0131
	IR	.4089 / .1595	.2039 / .1826	.3282 / .2954	-.0002 / -.0001	.0004 / .0004

Tab. 6.4.: Short-term **subjectivity** measurement (devel / test) for **arousal** (VGGish): The variance of true confidence σ^2_{true} is $.0277/.0248$.

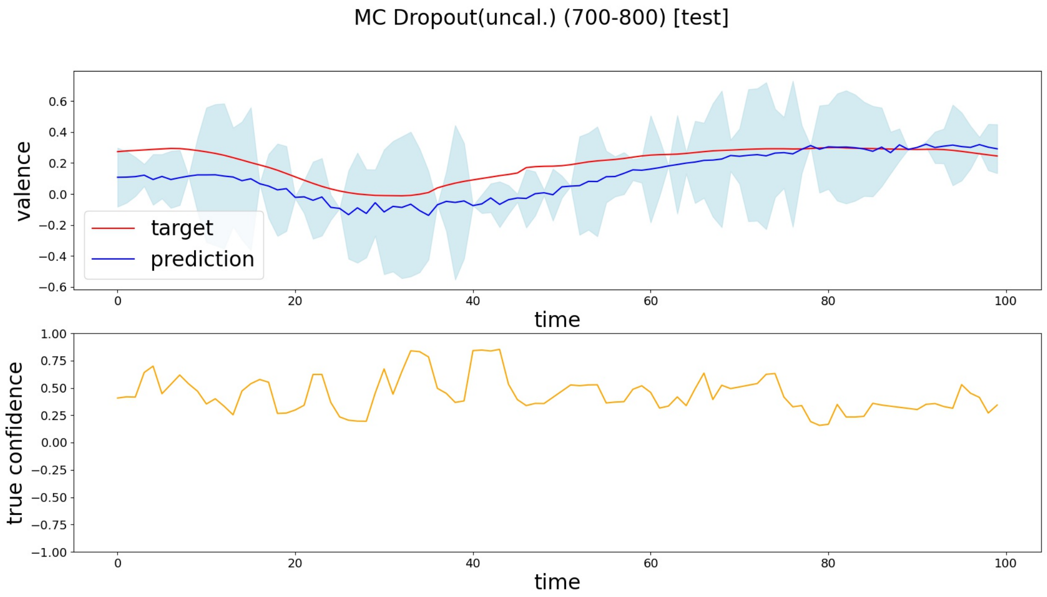

Fig. 6.2.: This plot visualizes **short-term uncertainty quantification**. The upper graph shows the label (red), the models prediction (blue) and its quantified uncertainty $\hat{U} \in [-1, 1]$ (light blue). To obtain expressive visualizations for $\hat{U}$, we compute $\frac{|\hat{U}-1|}{2}$. By this, the light blue area becomes large if the original quantified correlations are small (so uncertainty is large, respectively). The larger the light blue area, the less the model's confidence in its prediction. In this particular case, uncertainty is quantified with Monte Carlo Dropout.
Further, the yellow line represents true (or at least expected) uncertainty which (here) is defined as the average **PCC between multiple annotations**, as given with equation 4.4. A correlation of $+1$ means that all annotators perfectly agree (zero subjectivity/total objectivity) about the sample's annotation, as while -1 means negative correlation, so they absolutely disagree (total subjectivity/zero objectivity). What we expect (or at least hope) to observe is that the model's uncertainty correlates with the subjectivity of the annotation. This would mean that we observe larger light blue areas for smaller yellow values.
The other approach to define true uncertainty is the **PCC between target and prediction**, so the predictive performance (eq. 4.5).

emo. dim.	criterion	CCC$_{emo}$	MSE	CCC	σ^2_{true}	σ^2_{pred}
valence	CCC	.4006 / .5916	.0558 / .0385	-.0049 / .0060	.0272 / .0279	.0109 / .0108
	MSE	.4527 / .5778	.0392 / .0290	-.0119 / -.0054	.0272 / .0279	.0003 / .0002
arousal	CCC	.3566 / .2414	.1227 / .0428	-.0469 / -.0527	.0277 / .0248	.0123 / .0153
	MSE	.3300 / .2634	.0914 / .0254	-.0181 / -.0020	.0277 / .0248	.0009 / .0006

Tab. 6.5.: This table contains the results (devel / test) of directly predicting the subjectivity among annotations with an additional output node. We repeate the experiments with CCC loss and Mean Squared Error (MSE) (defined by $1/N \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$) as loss function and use both of them to evaluate the performance of the prediction either way (actually CCC, not CCC loss for evaluation). σ^2_{true} denotes the variance of the label y and σ^2_{pred} the variance of the model's prediction for subjectivity $\hat{y}$. Additionally, the CCC score for the prediction of the emotional state CCC$_{emo}$ is given. The node that generates the output which scores CCC$_{emo}$ (prediction of the emotional state) is optimized using CCC loss (eq. 4.11) as scoring rule, no matter if the criterion for scoring subjectivity was CCC loss or MSE. We repeat these experiments with stopping convergence of the initial output node for the emotional state after some times, to confirm that the inability of the model to predict subjectivity does not come from overruling the loss by the initial node. But also that change did not yield different results.

failure of subjectivity among raters as true confidence (tables 6.3 and 6.4). To confirm the impossibility of (locally) quantifying uncertainty that represents the actual uncertainty based on subjectivity among annotations, we additionally try to obtain just these subjectivity without detours, namely by straightly predicting it with an extra output node. The results in table 6.5 show that subjectivity can not be predicted with a NN. So we have to assume that the features do not contain information about subjectivity among annotations (as defined by now). This is not what we expect[2], but it might explain the poor results of attempts to deduct the subjectivity from predicted uncertainty.

On a conceptual level, it indeed does make sense that short-term predictions and confidence quantification fail. Three time steps correspond to $3/4$ seconds (compare section 5.1) which is certainly way less than the reaction time of a human annotator.

6.2 Global Subjectivity and Global Uncertainty

The last section discloses issues with measurements of emotion and uncertainty per time step. Hence, we decide to look at uncertainty, like we look at emotion recognition for training and convergence, namely globally (per whole example, respectively). This section shows that uncertainty can be measured implicitly. But still, subjectivity among annotations is not the (main) cause for it.

[2] We assume that subjectivity among those annotations, that later on forms the additional label, which our model is trained on, would be a main driver for the model's uncertainty.

method	cal.	gENCE	MAE	CCC	σ^2_{pred}
Ensemble [BM]	-	.0885 / .0369	.2317 / .0926	-.0030 / -.0392	.0472 / .0068
Ensemble	-	.0740 / .0753	.2361 / .2481	**.1766 / .2688**	.0444 / .0520
	IR	.0000 / .0105	.1252 / .1491	.1271 / .1601	.0032 / .0044
MCD [BM]	-	.083 / .0960	.2175 / .2566	-.0028 / .0068	.0411 / .0543
MCD	-	.1053 / .1075	.2751 / .2832	.1766 / .2406	.0711 / .0745
	IR	.0020 / .0145	.1146 / .1382	.1632 / .1735	.0038 / .0043

Tab. 6.6.: Global **predictive performance** measurement (devel / test) for **valence** (BERT): The variances of true confidence σ^2_{true} are $.0470/.0642$ for Ensemble Averaging and $.0415/.0566$ for MCD.

method	cal.	gENCE	MAE	CCC	σ^2_{pred}
Ensemble [BM]	-	.0909 / .0731	.2407 / .1879	-.0037 / -.0172	.0495 / .0313
Ensemble	-	.0769 / .0636	.2420 / .2115	**.2336 / .1224**	.0513 / .0309
	IR	.0000 / .0049	.1186 / .0915	.2217 / .1083	.0063 / .0028
MCD [BM]	-	.0818 / .0656	.2144 / .1702	.0224 / -.0024	.0416 / .0264
MCD	-	.1183 / .1164	.2904 / .3036	.2049 / .0506	.0894 / .0698
	IR	.0021 / .0079	.1033 / .0898	.2673 / .0840	.0065 / .0036

Tab. 6.7.: Global **predictive performance** measurement (devel / test) for **arousal** (VGGish): The variances of true confidence σ^2_{true} are $.0506/.0325$ for Ensemble Averaging and $.0399/.0271$ for MCD.

Tables 6.6 and 6.7 show the results for uncertainty, defined as the predictive performance of the model (4.16). For both, the CCC is above zero. Although the correlations are far from being highly correlated (CCC $\rightarrow 1$), they seem to exist. Further, we observe that calibration leads to a improvement in terms of MAE, but a decline of σ^2_{pred}. This indicates that calibration is still difficult for the calibrator as it tends to predict the mean of the target space. But σ^2_{pred} does not become zero and still CCC > 0, so calibration is possible, at least for some samples of the uncalibrated measurements of confidence. This observations are confirmed by visualizing the correlation between predicted and true uncertainty with Reliability Diagrams (sec. A.1). Each (blue) Reliability Diagram for predictive performance shows that predicted confidence (x-axis) correlates with true confidence (y-axis). It seems that calibration leads to concentration on the mean. But nevertheless, the mapping, learned by the calibrator, still holds on the test set. Further, we can observe that quality of predicted uncertainty, in terms of predictive performance, is related to the model's performance of predicting the emotional state. The model predicts valence more accurate on the test set than on the validation data (table 6.1 for CCC$_{emo}$). This reflects on its ability of quantifying its predictive performance, represented by uncertainty (table 6.6). It behaves contrarily for arousal. Emotional state and predictive performance perform better on the validation data (table 6.2 for CCC$_{emo}$ and table 6.7). This is another indicator for the link between quantified uncertainty and predictive performance.

It looks different for subjectivity as representation for real uncertainty (eq. 4.15). Although CCCs are slightly above those of the benchmarks (≈ 0), they're far smaller

method	cal.	gENCE	MAE	CCC	σ^2_{pred}
Ensemble [BM]	-	.0359 / .0373	.0935 / .0913	.0004 / .0000	.0076 / .0072
Ensemble	-	.0873 / .1014	.2266 / .2509	.0349 / .0438	.0444 / .0520
	IR	.0000 / .0089	.0610 / .0631	.0245 / .0309	.0001 / .0001
MCD[BM]	-	.0365 / .0361	.0939 / .0929	-.0199 / -.0149	.0076 / .0072
MCD	-	.1178 / .1295	.2807 / .2974	.0302 / .0373	.0711 / .0745
	IR	.0001 / .0093	.0610 / .0631	.0192 / .0299	.0001 / .0001

Tab. 6.8.: Global **subjectivity** measurement (devel / test) for **valence** (BERT): The variance of true confidence σ^2_{true} is $.0077/.0070$.

method	cal.	gENCE	MAE	CCC	σ^2_{pred}
Ensemble [BM]	-	.0369 / .0270	.0940 / .0701	-.0225 / -.0274	.0073 / .0041
Ensemble	-	.0905 / .0681	.2256 / .1823	.0767 / .0390	.0513 / .0309
	IR	.0000 / .0102	.0641 / .0551	**.1253 / .0565**	.0005 / .0002
MCD [BM]	-	.0356 / .0265	.0911 / .0694	.0112 / .0101	.0070 / 0041
MCD	-	.1330 / .1199	.2958 / .2857	.0402 / .0134	.0894 / .0698
	IR	.0001 / .0117	.0650 / .0569	.0574 / .0248	.0003 / .0001

Tab. 6.9.: Global **subjectivity** measurement (devel / test) for **arousal** (VGGish): : The variance of true confidence σ^2_{true} is $.0072/.0041$.

than for predictive performance as real confidence (compare tables 6.8 and 6.9). The initially assumed triangular equality of predictive performance, subjectivity among annotations and model's uncertainty seems not to hold, or at least to be less weak than expected. As the correlation between predictive performance and confidence is measurable, conceptual issues, like it has been the case for short-term measurements, should not be the reason for this results. We rather believe that there are multiple reasons that cause uncertainty and subjectivity among raters is just one (maybe even less important). We come back to that and other reasons that might have impact on our model's uncertainty in the conclusion (sec. 7.1).

We also present the CCC score between subjectivity and predictive performance (table 6.10) to confirm that they are weakly correlated (which is what we observe implicitly). Indeed, these results confirm, what we expect after investigating table 6.9. For the validation set and arousal, subjectivity and predictive performance are actually slightly correlated. This becomes visible (especially) for Ensemble Averaging within its Reliability Diagram A.7.

emo. dim.	method	CCC
valence	MCD	.0975 / .0970
	Ensemble	.0966 / 0945
arousal	MCD	.1347 / .0733
	Ensemble	.1227 / .0813

Tab. 6.10.: CCC (devel / test) between subjectivity and predictive performance as defined in equation 4.15 and 4.16 respectively.

Conclusion

7

Predictive uncertainty is crucial for any application. We proposed techniques for transferring the procedure of uncertainty quantification to contiguous predictions through replacing variance by correlation. Furthermore, we extended the measurement of true uncertainty, which is used to judge quality of predicted uncertainty, by another approach: we defined subjectivity among multiple (human) raters as additional comparative value for a model's confidence, next to the predictive performance of the model. This allowed us to investigate our initial assumption: predictive uncertainty is represented by the model's prediction error and driven by subjectivity among raters. As our experiments on local uncertainty measurement (sec. 6.1) turned out to fail because of conceptional issues that prevented successful uncertainty measurements, we restrict further considerations and thoughts on answering our research questions (sec. 1.2) to results from global uncertainties (sec. 6.2).

7.1 Discussion

We observed that there indeed seems to exist a coherence between predictive performance and predicted uncertainty (first research question). This is not that surprising as predictive performance is the common definition for true confidence. Actually, the correlations of about 0.2 (sec. 6.2) could even be considered as weak under the circumstance that the predictive performance, as measurement of true confidence, unites each source of uncertainty. No matter what underlying reasons (noise, lacking model knowledge, ...) evoke model's uncertainty, they should reflect on its predictive performance as well. We believe that this weaker than expected correlation is caused by multiple conceptional obstacles in our experiments, in contrast to basic single point regression. Particularly, we had to specify a fixed window length for the subsamples (sec. 4.2). The choice of $\pi = 20$ was, although considered, nevertheless an arbitrary guess. We introduced π as we wanted to measure uncertainty for multiple time steps together but also needed to be aware of varying uncertainties within a raw sample from the data set. But confidences do not vary in intervals of any fixed length. So we must admit that the shift from single point regression and variance to contiguous time series forecasting and correlation might induce (at least for our approach) vagueness in the measurements. Anyway, we have to recognize that we did not predict the model's uncertainty explicitly. Rather, we measured it implicitly from multiple predictions which are sequences of predictions themselves and differ because of randomness.[1] This causes vagueness. So, we should not expect perfectly calibrated and correlating uncertainties.

Regarding the coherence between subjectivity among annotations and predicted uncertainty, we observed that it is (if at all) barely available (second research question). This might indicate that subjectivity among annotations is only one source of uncertainty among others. *But what other sources?* (Sub-) samples, showing high disagreement between raters, (probably) do so because the sample was hard to annotate based on the features. This must have been the case because noise in the features predominates the signal or the signal does not transport expressive information at all. The sample is noisy. From this we derive that subjectivity among annotations indicates aleatory uncertainty. As mentioned yet, general predictive uncertainty puts together out of aleatory and epistemic uncertainty. Hence, the weaker correlations between subjectivity among annotations and predicted uncertainty, than those between predictive performance and predicted uncertainty, might be explained by the disregard of epistemic uncertainty in this definition of true uncertainty. Epistemic uncertainty is caused by lack of knowledge from the model about the fed in data. These contexts are not included into subjectivity among raters as definition for true confidence. Furthermore, we have to note that subjectivity among raters does not fully represent aleatory uncertainty and that it is just an approximation for it.

7.2 Future Work

As described in the last section, the choice of $\pi = 20$ was arbitrary and one could vary it in further experiments to check if there exists a length that minimizes the intersection between different levels of confidence inside sub-samples. Further, there is no need that the sub-samples do not overlap. Overlapping would increase the amount of sub-samples and, as a result, may allow more expressive evaluation. Overlapping sub-samples would be conceptually similar to our local uncertainty approaches (sec. 4.1) with consideration of the latest $t = \pi$ time steps (instead of $t = 3$ and with $t = \pi >> 3$) at each point. Also, we could investigate the capability of tCCC to quantify confidence by parameterizing it with (much) larger values for w_l and w_u than before, and applying it (globally) with non-overlapping sub-samples or (locally) with large t.

[1] Random weight initialization for seeds at Ensemble Averaging and randomly switched off nodes during inference for Monte Carlo Dropout

Epistemic uncertainty appears if the model does not understand the features which are given to it. To confirm the impact on epistemic uncertainty on predicted uncertainty, as claimed in the last section, we could use simpler (actually less powerful) feature embedding techniques and try to observe if this increases predicted uncertainty and decreases the proportion of aleatory uncertainty relative to the overall uncertainty. Furthermore, we could think about manually quantifying the lack of knowledge for a sub-sample[2] and use this as a definition for true uncertainty, as we did with subjectivity among annotations.

For obtaining the final prediction with Monte Carlo Dropout and Ensemble Averaging, we used the mean of multiple predictions. One could use advanced fusion techniques, like the Evaluator Weighted Estimator[3] [47] instead of averaging, to obtain the final prediction. This change would also reflect on predictive performance and make it, as a definition of true confidence, more reliable.

[2] For instance, by calculating the number of words that are unknown to the BERT tokenizer. A tokenizer translates real words into numbers (or tokens) so that the BERT model can process them.

[3] It generates a kind of a *mean series* from multiple individual series but with outliers ignored so that they do not distort the overall picture.

List of Figures

List of Tables

Acronyms

BERT	Bidirectional Encoder Representations from Transformers
BM	Benchmark
CCC	Concordance Correlation Coefficient
DL	Deep Learning
DPN	Dirichlet Prior Network
ECE	Expected Calibration Error
ENCE	Expected Normalized Calibration Error
gENCE	General Uncertainty ENCE
IR	Isotonic Regression
KL	Kullback–Leibler
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCD	Monte Carlo Dropout
MSE	Mean Squared Error
NLL	Negative log-likelihood
NN	Neural Network
OOD	out-of-distribution
PCC	Pearson Correlation Coefficient
PDF	Probability Density Function
RNN	Recurrent Neural Network
tCCC	tilted CCC

Appendix A

A.1 Reliability Diagrams

To measure the quality of calibration for global uncertainty measurement (sec. 6.2), we apply Reliability Diagrams, similar to those presented in section 3.2.1. Just like at figure 3.6, the x-axis represents true confidence and the y-axis predicted confidence. But in our case, both axis range from -1 (totally negative correlation) to $+1$ (perfectly correlated). We use the same binning logic as for gENCE. Predicted uncertainty gets binned to one of 5 bins and each bin is located on the mean of its samples. If no sample falls into a certain bin, the bin is not shown in the diagram. Furthermore, a bin becomes more transparent, the less confidence predictions it has assigned relatively. The orange diagonal points out the height a bin should have to be perfectly correlated. One figure contains the Reliability Diagrams for one approach and one emotional dimension, so plots for uncalibrated prediction and calibrated ones for both data subsets (validation and test) each.

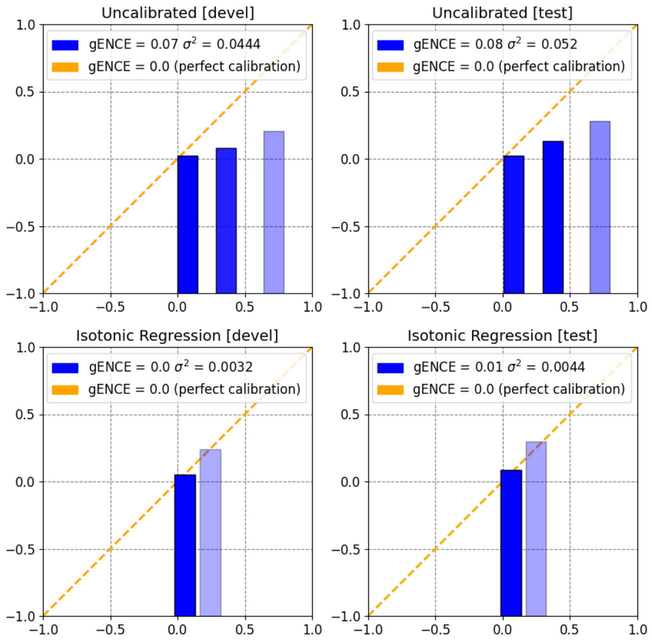

Fig. A.1.: Reliability diagram: **Ensemble Averaging** for **valence** with **Predictive Performance** as true uncertainty.

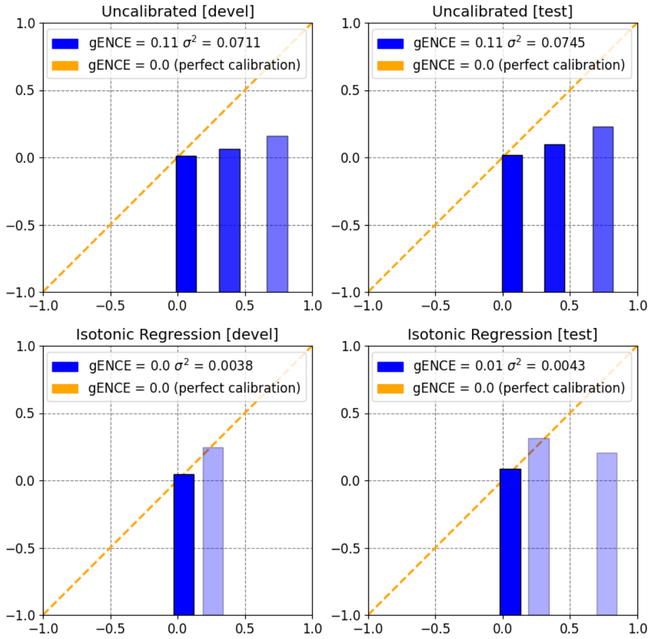

Fig. A.2.: Reliability diagram: **Monte Carlo Dropout** for **valence** with **Predictive Performance** as true uncertainty.

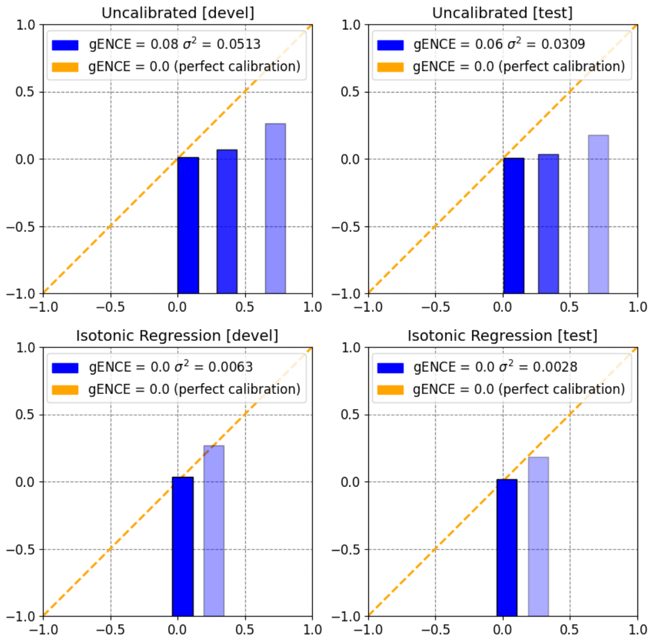

Fig. A.3.: Reliability diagram: **Ensemble Averaging** for **arousal** with **Predictive Performance** as true uncertainty.

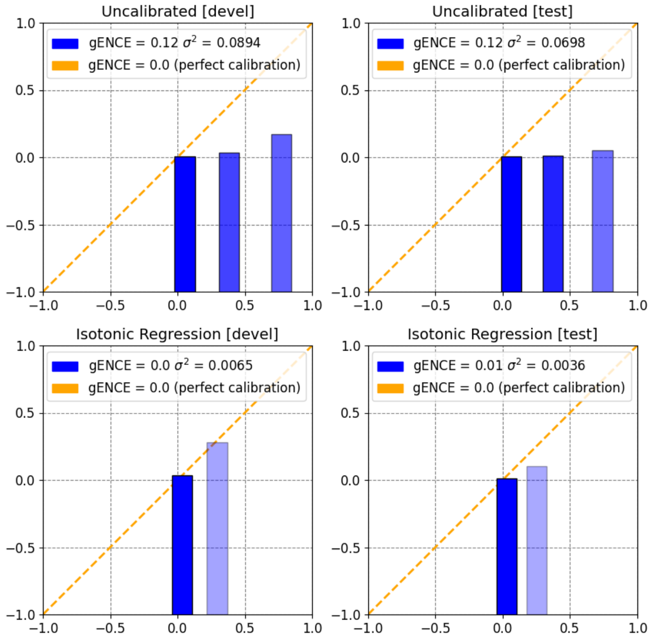

Fig. A.4.: Reliability diagram: **Monte Carlo Dropout** for **arousal** with **Predictive Performance** as true uncertainty.

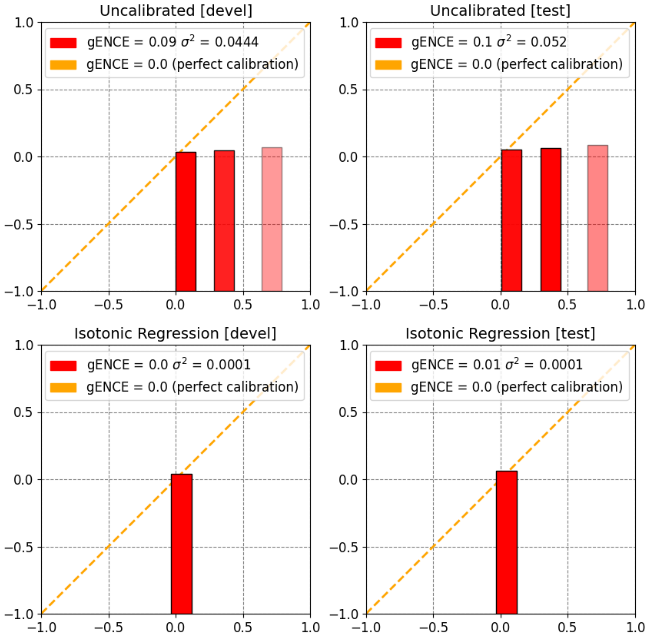

Fig. A.5.: Reliability diagram: **Ensemble Averaging** for **valence** with **Subjectivity among raters** as true uncertainty.

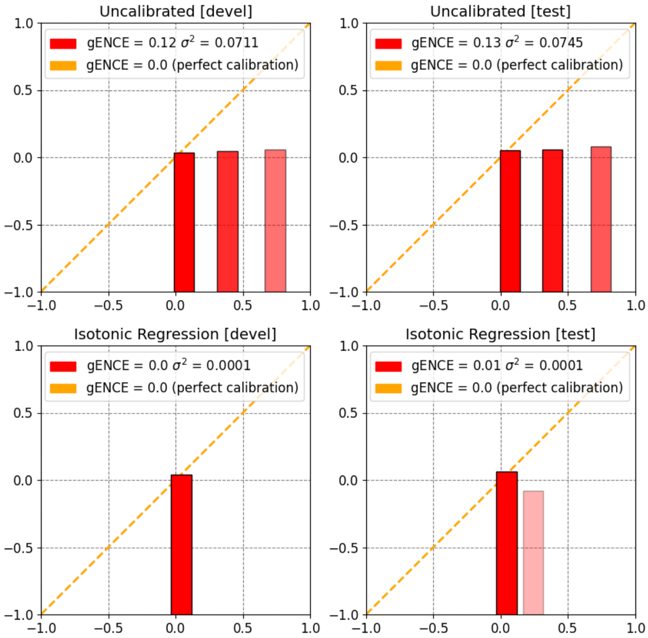

Fig. A.6.: Reliability diagram: **Monte Carlo Dropout** for **valence** with **Subjectivity among raters** as true uncertainty.

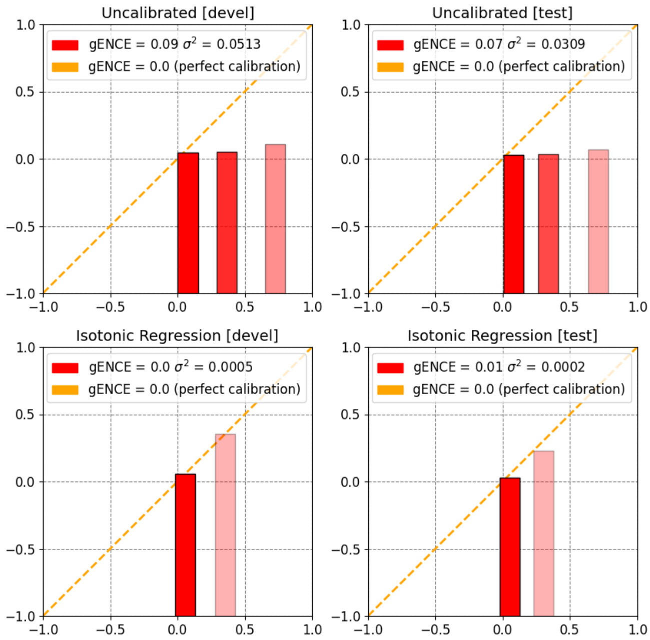

Fig. A.7.: Reliability diagram: **Ensemble Averaging** for **arousal** with **Subjectivity among raters** as true uncertainty.

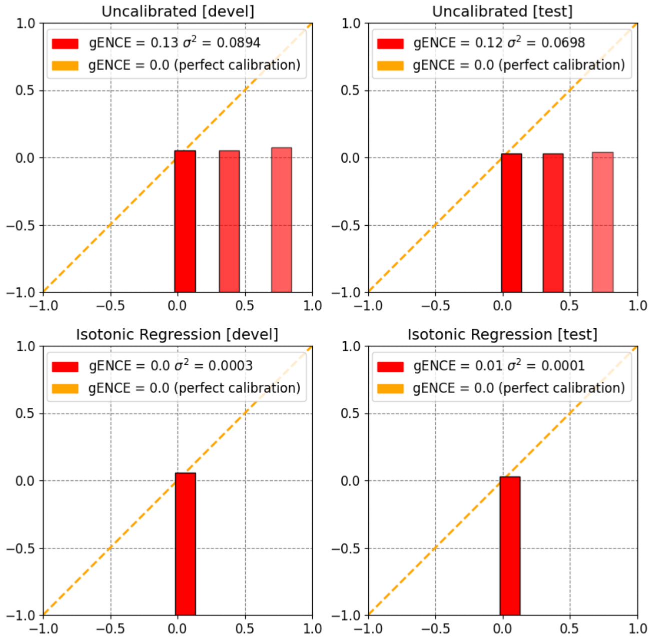

Fig. A.8.: Reliability diagram: **Monte Carlo Dropout** for **arousal** with **Subjectivity among raters** as true uncertainty.

Bibliography

[1] Lukas Stappen, Alice Baird, Lukas Christ, et al. “The MuSe 2021 Multimodal Sentiment Analysis Challenge: Sentiment, Emotion, Physiological-Emotion, and Stress”. In: *Proceedings of the 2nd International Multimodal Sentiment Analysis Challenge and Workshop* (2021).

[2] Alex Kendall and Yarin Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Advances in Neural Information Processing Systems* (2017).

[3] Georgios Rizos and Björn W. Schuller. “Average Jane, Where Art Thou? – Recent Avenues in Efficient Machine Learning Under Subjectivity Uncertainty”. In: *Information Processing and Management of Uncertainty in Knowledge-Based Systems* (2020).

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[5] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. “Understanding of a convolutional neural network”. In: *International Conference on Engineering and Technology* (2017).

[6] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* (1997).

[7] Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019).

[9] Shawn Hershey, S. Chaudhuri, D. Ellis, et al. “CNN architectures for large-scale audio classification”. In: *International Conference on Acoustics, Speech and Signal Processing* (2017).

[10] James Russell. “A Circumplex Model of Affect”. In: *Journal of Personality and Social Psychology* (1980).

[11] Yi-Hsuan Yang and Homer Chen. “Machine Recognition of Music Emotion: A Review”. In: *Association for Computing Machinery Transactions on Intelligent Systems and Technology* (2012).

[12] Atsushi Ando, Satoshi Kobashikawa, Hosana Kamiyama, et al. “Soft-Target Training with Ambiguous Emotional Utterances for DNN-Based Speech Emotion Classification”. In: *International Conference on Acoustics, Speech and Signal Processing* (2018).

[13] Jing Han, Zixing Zhang, Zhao Ren, and Björn Schuller. “Exploring Perception Uncertainty for Emotion Recognition in Dyadic Conversation and Music Listening”. In: *Cognitive Computation* (2020).

[14] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning* (2016).

[15] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. “Weight Uncertainty in Neural Networks”. In: *Proceedings of Machine Learning Research* (2015).

[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017).

[17] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. “Hyperparameter Ensembles for Robustness and Uncertainty Quantification”. In: *Advances in Neural Information Processing Systems* (2020).

[18] Roger Koenker. *Quantile Regression*. 2005.

[19] Omer Achrack, Ouriel Barzilay, and Raizy Kellerman. *Multi-Loss Sub-Ensembles for Accurate Classification with Uncertainty Estimation*. 2020.

[20] Javier Antoran, James Allingham, and José Miguel Hernández-Lobato. “Depth Uncertainty in Neural Networks”. In: *Advances in Neural Information Processing Systems* (2020).

[21] Yeming Wen, Dustin Tran, and Jimmy Ba. “BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning”. In: *International Conference on Learning Representations* (2020).

[22] Andrey Malinin and Mark Gales. “Predictive Uncertainty Estimation via Prior Networks”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018).

[23] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. “Ensemble Distribution Distillation”. In: *International Conference on Learning Representations* (2020).

[24] Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales. “Regression Prior Networks”. In: *arXiv e-prints* (2020).

[25] Murat Sensoy, Lance Kaplan, and Melih Kandemir. “Evidential Deep Learning to Quantify Classification Uncertainty”. In: *Advances in Neural Information Processing Systems* (2018).

[26] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. “Deep evidential regression”. In: *Advances in Neural Information Processing Systems* (2020).

[27] Daniel Zügner Charpentier Bertrand and Stephan Günnemann. “Posterior Network: Uncertainty Estimation without OOD Samples via Density-Based Pseudo-Counts”. In: *Advances in Neural Information Processing Systems* (2020).

[28] Danilo Rezende and Shakir Mohamed. “Variational Inference with Normalizing Flows”. In: *Proceedings of Machine Learning Research* (2015).

[29] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. “Normalizing Flows for Probabilistic Modeling and Inference”. In: *Journal of Machine Learning Research* (2021).

[30] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. “On calibration of modern neural networks”. In: *International Conference on Machine Learning* (2017).

[31] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2009.

[32] Glenn Brier. “Verification of forecasts expressed in terms of probability”. In: *Monthly Weather Review* (1950).

[33] Max-Heinrich Laves, Sontje Ihler, Karl-Philipp Kortmann, and Tobias Ortmaier. *Uncertainty Calibration Error: A New Metric for Multi-Class Classification*. 2021.

[34] Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Linchuan Zhang, and Dustin Tran. “Measuring Calibration in Deep Learning”. In: *arXiv e-prints* (2020).

[35] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. “Accurate Uncertainties for Deep Learning Using Calibrated Regression”. In: *Proceedings of Machine Learning Research* (2018).

[36] Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. “Evaluating and Calibrating Uncertainty Prediction in Regression Tasks”. In: *arXiv e-prints* (2020).

[37] John Platt. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”. In: *Advances Large Margin Classifiers* (1999).

[38] Philippe Xu, Franck Davoine, and Thierry Denœux. “Evidential multinomial logistic regression for multiclass classifier calibration”. In: *18th International Conference on Information Fusion* (2015).

[39] Meelis Kull, Telmo Silva Filho, and Peter Flach. “Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (2017).

[40] Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. “Distribution calibration for regression”. In: *Proceedings of the 36th International Conference on Machine Learning* (2019).

[41] Ananya Kumar, Percy S Liang, and Tengyu Ma. “Verified Uncertainty Calibration”. In: *Advances in Neural Information Processing Systems* (2019).

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* (2011).

[43] Jayaraman J. Thiagarajan, Bindya Venkatesh, Prasanna Sattigeri, and Peer-Timo Bremer. “Building Calibrated Deep Models via Uncertainty Matching with Auxiliary Interval Predictors”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2020).

[44] Ranganath Krishnan and Omesh Tickoo. “Improving model calibration with accuracy versus uncertainty optimization”. In: *Advances in Neural Information Processing Systems* (2020).

[45] Michel Valstar, Jonathan Gratch, Bjorn Schuller, et al. “AVEC 2016 - Depression, Mood, and Emotion Recognition Workshop and Challenge”. In: *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge* (2016).

[46] Licai Sun, Zheng Lian, Jianhua Tao, Bin Liu, and Mingyue Niu. “Multi-Modal Continuous Dimensional Emotion Recognition Using Recurrent Neural Network and Self-Attention Mechanism”. In: *Proceedings of the 1st International on Multimodal Sentiment Analysis in Real-Life Media Challenge and Workshop* (2020).

[47] Michael Grimm and Kristian Kroschel. “Evaluation of natural emotions using self assessment manikins”. In: *Proceedings of Automatic Speech Recognition and Understanding Workshop* (2005).