VARIATIONAL GAUSSIAN PROCESS DATA UNCERTAINTY

Jeremy H. M. Wong, Huayun Zhang, and Nancy F. Chen

Institute for Infocomm Research (I^2R), A*STAR, Singapore

ABSTRACT

A Gaussian process (GP) computes an explainable distributional uncertainty by hypothesising higher output uncertainty for query inputs far from the training inputs. However, a GP may not capture data uncertainty well. Accurate data uncertainty estimation may be important for subjective tasks, such as Spoken Language Assessment (SLA), where human expert raters may disagree on the output scores. This paper shows that a variational approximation of a GP has capacity to learn data uncertainty from the training data. However, standard training criteria tune only a scalar noise hyper-parameter toward the standard deviation of the output reference, thereby limiting the learning of this uncertainty. A training criterion is proposed to explicitly encourage the GP posterior to emulate the distribution of scores from multiple raters. Experiments on the speechocean762 SLA task show that this allows the GP to better express data uncertainty and improves the modelling of inter-rater disagreements.

***Index Terms*—** Data uncertainty, Gaussian process, variational, Kullback-Leibler divergence, spoken language assessment

1. INTRODUCTION

The Gaussian Process (GP) [1] has a different modelling paradigm than a Neural Network (NN). A NN stores information about the training data in its many parameters, then uses these learned parameters to process run-time queries. During run-time, the GP instead compares the similarity between the query input and all training inputs, to hypothesise an output that is correlated with the reference outputs of nearby training inputs. This yields an explainable distributional uncertainty [2], where outputs for query inputs that reside far from the training inputs are hypothesised with high uncertainty. NNs may instead be overly confident about wrong hypotheses, especially for out-of-domain query inputs [3]. A GP may also reduce the influence of model uncertainty, through its interpretation as being a marginalisation over functions [1]. This is the uncertainty of not knowing which model design is most appropriate for the task [4]. However, the standard GP does not naturally compute data uncertainty [5], which is the uncertainty about what the output should be, due to noise or insufficient information in the input, or due to overlapping inputs from different output classes [4]. This paper aims to improve the ability of the GP to compute data uncertainty, thereby allowing this model type to capture a more holistic representation of all three forms of uncertainties. It is shown that by using a variational approximation of the GP, the variational parameters may introduce the ability to capture data uncertainty from the training data. However, it is also shown that standard approaches to train standard and variational GPs may not yield hyper-parameter and variational parameter values that capture data uncertainty. This paper then proposes a training criterion to allow for data uncertainty to be learned.

This work was supported by the A*STAR Computational Resource Centre through the use of its high performance computing facilities.

The proposed approaches are evaluated on a Spoken Language Assessment (SLA) task. In automatic SLA, a model hypothesises a score from input speech, reflecting an aspect of the speaker's oral proficiency. Examples of oral proficiency aspects include pronunciation accuracy, intonation, fluency, prosody, sentence completeness, task completion, and topic relevance. The hypothesised score is expected to emulate one that an expert rater would have assigned. However, SLA is a subjective task, where different expert raters often assign different scores to the same input speech. These inter-rater disagreements are reminiscent of data uncertainty, as each rater assigns the output based on only the available input information. The subjectivity arises because of the difficulty of writing a scoring rubric that covers all possible disagreement scenarios. This limited rubric coverage leaves room for the rater's subjective bias to influence the score assigned. Increasing the coverage of the rubric may reduce the inter-rater disagreements, but may then not generalise well to the needs of a new application scenario, where different rubric choices may instead be preferred. Instead of seeking to reduce inter-rater disagreements, this paper aims to improve the model's ability to capture such data uncertainty. A model that accurately predicts rater disagreement provides useful information that can better inform the feedback given by the system. This can mitigate poor user experience by not overly penalising a user for a low hypothesised score, when it is also predicted that raters would likely disagree. When rater disagreement is predicted, the system can be designed to seek user clarification or expert human intervention.

2. RELATED WORK

The extensions to a GP investigated here consider tasks where multiple reference output samples per input are provided for the same task. This differs from a multi-output GP [6, 7], which instead considers a multi-task framework [8], where the multiple outputs represent different tasks. The availability of multiple reference output samples is particularly relevant to subjective tasks, where annotators may not agree on what the correct output should be for each input. This paper evaluates the proposed approach on an SLA task. Other subjective tasks that are often labelled with multiple output samples include emotion recognition [9] and stutter detection [10]. The proposed approach trains a GP to emulate the density function represented by the multiple reference output samples. A NN can also be trained toward a reference distribution [11]. Knowledge distillation [12, 13] instead trains a student model to emulate an output distribution obtained from a teacher model. The proposal aims for a model to emulate the uncertainty expressed between multiple human raters, assuming that it is difficult to obtain a true reference score in a subjective task. Rasch modelling [14] and proportional reduction mean squared error [15] instead assume an observational variance that occludes a true reference score, and aim to infer this true reference from the multiple rater scores. In this paper, a single model both infers a hypothesis and expresses uncertainty about that hypothesis. It

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

is also possible to compute the confidence using a separately trained confidence estimation module [16].

3. GAUSSIAN PROCESS

This section describes the formulation of a standard GP. Given a set of input features, $\mathbf{X}$, a GP computes latent variables, $\boldsymbol{f}$, that are jointly Gaussian distributed in the GP prior,

$$p\left(\boldsymbol{f}|\mathbf{X}\right) = \mathcal{N}\left(\boldsymbol{f}; \mathbf{0}, \mathbf{K}^{\mathbf{XX}}\right). \tag{1}$$

The covariance between the latent variables is computed using the kernel, $\mathbf{K}$, which measures a similarity between the input features. For brevity, the short-form notation of $\mathbf{K}^{\mathbf{XX}'}$ is used to represent a kernel that measures the similarity between features $\mathbf{X}$ and $\mathbf{X}'$. The experiments in this paper use the squared exponential kernel,

$$k_{ij}^{\mathbf{XX}'} = s^2 \exp\left[-\frac{\left(\boldsymbol{x}_i - \boldsymbol{x}'_j\right)^\top \left(\boldsymbol{x}_i - \boldsymbol{x}'_j\right)}{2l^2}\right], \tag{2}$$

where s and l are the scale and length hyper-parameters respectively, and i and j are data point indexes. When given the latent variables, the outputs, $\boldsymbol{y}$, are assumed to be Gaussian distributed,

$$p\left(\boldsymbol{y}|\boldsymbol{f}\right) = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{f}, \sigma^2\mathbf{I}\right), \tag{3}$$

where hyper-parameter σ expresses noise in the observed output. A GP assumes that the outputs are conditionally independent of the inputs when given the latent variables, with a marginal likelihood of

$$p\left(\boldsymbol{y}|\mathbf{X}\right) = \int p\left(\boldsymbol{y}|\boldsymbol{f}\right) p\left(\boldsymbol{f}|\mathbf{X}\right) d\boldsymbol{f} = \mathcal{N}\left(\boldsymbol{y}; \mathbf{0}, \mathbf{K}^{\mathbf{XX}} + \sigma^2\mathbf{I}\right). \tag{4}$$

A standard approach to tune the GP hyper-parameters is to maximise the Marginal log-Likelihood (ML) of the training data,

$$\mathcal{F}_{\text{ML}} = \log p\left(\boldsymbol{y}|\mathbf{X}\right). \tag{5}$$

For a query input, $\widehat{\boldsymbol{x}}$, a posterior over the query latent variable, $\widehat{f}$, is

$$p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X}\right) = \frac{p\left(\widehat{f}, \boldsymbol{y}\middle|\widehat{\boldsymbol{x}}, \mathbf{X}\right)}{p\left(\boldsymbol{y}|\mathbf{X}\right)} = \mathcal{N}\left(\widehat{f}; \widehat{\mu}, \widehat{v}\right), \tag{6}$$

where

$$\widehat{\mu} = \boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}\top}\left[\mathbf{K}^{\mathbf{XX}} + \sigma^2\mathbf{I}\right]^{-1}\boldsymbol{y} \tag{7}$$

$$\widehat{v} = k^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}} - \boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}\top}\left[\mathbf{K}^{\mathbf{XX}} + \sigma^2\mathbf{I}\right]^{-1}\boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}}. \tag{8}$$

The output for this query is then inferred from the output posterior,

$$p(\widehat{y}|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X}) = \int p\left(\widehat{y}\middle|\widehat{f}\right) p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X}\right) d\widehat{f} = \mathcal{N}(\widehat{y}; \widehat{\mu}, \widehat{v} + \sigma^2). \tag{9}$$

4. MULTIPLE OUTPUT SAMPLES

The standard GP assumes that the training data comprises only a single scalar reference output for each input. This section describes a proposed extension to allow the GP to utilise multiple reference output samples in the training data. In SLA, a scalar reference score can be computed as either the majority vote, average, or median of the multiple rater scores [17]. This scalar reference can then be used with a standard GP [18]. However, compressing the multiple reference outputs into a single scalar reference may omit information about data uncertainty in the training set. A naive method of allowing the GP to use multiple reference output samples is to replicate the training input for each reference output sample associated with that data point. However, this increases the size of the kernel, which then necessitates more expensive computation during both training and run-time inference. The computational cost can be reduced by noticing that the kernel computes perfectly correlated latent variables for repeated inputs. Perfectly correlated random variables will always have the same value as each other, and there is thus no need to separately compute each of them [19]. Without repeating the input and latent variable for each reference output sample, the output density can be expressed as

$$p\left(\overline{\mathbf{Y}}\middle|\boldsymbol{f}\right) = \prod_i \prod_{r=1}^{R_i} \mathcal{N}\left(\overline{y}_{ir}; f_i, \sigma^2\right) \tag{10}$$

$$= \mathcal{N}\left(\overline{\boldsymbol{\mu}}; \boldsymbol{f}, \mathbb{D}\left(\boldsymbol{\omega}\right)\right) \mathcal{N}\left(\overline{\boldsymbol{\eta}}'; \mathbf{0}, \mathbb{D}\left(\boldsymbol{\omega}\right)\right) g, \tag{11}$$

where

$$\overline{\mu}_i = \frac{1}{R_i}\sum_{r=1}^{R_i}\overline{y}_{ir} \quad \text{and} \quad \overline{\eta}'_i = \sqrt{\frac{1}{R_i}\sum_{r=1}^{R_i}\left(\overline{y}_{ir} - \overline{\mu}_i\right)^2}, \tag{12}$$

$$\omega_i = \frac{\sigma^2}{R_i} \quad \text{and} \quad g = \prod_i \frac{\left(2\pi\sigma^2\right)^{1-\frac{R_i}{2}}}{R_i}. \tag{13}$$

Here, $\overline{\boldsymbol{\mu}}$ and $\overline{\boldsymbol{\eta}}'$ are the mean and biased standard deviation of the multiple reference output samples, the operator $\mathbb{D}\left(\boldsymbol{\omega}\right)$ converts vector $\boldsymbol{\omega}$ into a diagonal matrix, and R_i is the number of raters who annotated the ith input. Also, $\overline{\mathbf{Y}}$ represents a collection of the reference outputs from all raters, where $\overline{y}_{ir}$ is the score assigned by the rth rater for the ith input. This output density can be combined with the GP prior in (1) to compute a joint marginal likelihood of

$$p\left(\overline{\mathbf{Y}}\middle|\mathbf{X}\right) = \int p\left(\overline{\mathbf{Y}}\middle|\boldsymbol{f}\right) p\left(\boldsymbol{f}|\mathbf{X}\right) d\boldsymbol{f} \tag{14}$$

$$= \mathcal{N}\left(\overline{\boldsymbol{\mu}}; \mathbf{0}, \mathbf{K}^{\mathbf{XX}} + \mathbb{D}\left(\boldsymbol{\omega}\right)\right) \mathcal{N}\left(\overline{\boldsymbol{\eta}}'; \mathbf{0}, \mathbb{D}\left(\boldsymbol{\omega}\right)\right) g. \tag{15}$$

The hyper-parameters can be tuned to the multiple reference output samples by maximising the Joint Marginal log-Likelihood (JML),

$$\mathcal{F}_{\text{JML}} = \log p\left(\overline{\mathbf{Y}}\middle|\mathbf{X}\right). \tag{16}$$

Inference can also be performed with the multiple reference output samples. The joint density between the query latent variable and the training set multiple output scores is

$$p\left(\widehat{f}, \overline{\mathbf{Y}}\middle|\widehat{\boldsymbol{x}}, \mathbf{X}\right) = \int p\left(\widehat{f}, \boldsymbol{f}\middle|\widehat{\boldsymbol{x}}, \mathbf{X}\right) p\left(\overline{\mathbf{Y}}\middle|\boldsymbol{f}\right) d\boldsymbol{f} \tag{17}$$

$$= \mathcal{N}\left(\begin{bmatrix}\widehat{f} \\ \overline{\boldsymbol{\mu}}\end{bmatrix}; \mathbf{0}, \begin{bmatrix}\mathbf{K}^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}} & \mathbf{K}^{\widehat{\boldsymbol{x}}\mathbf{X}} \\ \mathbf{K}^{\mathbf{X}\widehat{\boldsymbol{x}}} & \mathbf{K}^{\mathbf{XX}} + \mathbb{D}(\boldsymbol{\omega})\end{bmatrix}\right) \mathcal{N}\left(\overline{\boldsymbol{\eta}}'; \mathbf{0}, \mathbb{D}(\boldsymbol{\omega})\right) g.$$

The posterior of the query latent variable is then

$$p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}, \overline{\mathbf{Y}}, \mathbf{X}\right) = \frac{p\left(\widehat{f}, \overline{\mathbf{Y}}\middle|\widehat{\boldsymbol{x}}, \mathbf{X}\right)}{p\left(\overline{\mathbf{Y}}\middle|\mathbf{X}\right)} = \mathcal{N}\left(\widehat{f}; \breve{\mu}, \breve{v}\right), \tag{18}$$

where

$$\breve{\mu} = \boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}\top}\left[\mathbf{K}^{\mathbf{XX}} + \mathbb{D}\left(\boldsymbol{\omega}\right)\right]^{-1}\overline{\boldsymbol{\mu}} \tag{19}$$

$$\breve{v} = k^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}} - \boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}\top}\left[\mathbf{K}^{\mathbf{XX}} + \mathbb{D}\left(\boldsymbol{\omega}\right)\right]^{-1}\boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}}. \tag{20}$$

Finally, the query output posterior is

$$p\left(\widehat{y}\middle|\widehat{\boldsymbol{x}},\overline{\mathbf{Y}},\mathbf{X}\right)=\mathcal{N}\left(\widehat{y};\breve{\mu},\breve{v}+\sigma^2\right). \tag{21}$$

This formulation is referred to as the *joint GP*.

5. PILOT STUDY ON SIMULATED DATA

The concepts discussed in sections 6 and 7 are visualised through a toy example in Fig. 1. This simulated dataset comprises a one-dimensional sine wave, with missing input regions and noise in the output. The training inputs comprise 120 equally spaced points from 10 to 40 and another 120 points from 60 to 90. The input regions before 10, between 40 to 60, and after 90 simulate a lack of training data coverage. For input regions between 10 to 20, 30 to 40, 60 to 70, and 80 to 90, the reference output is assigned as the sine of the input, simulating noise-free outputs. Within input region 20 to 30, a single reference output is sampled per input from a normal density function with a mean at the sine of the input, simulating Gaussian noise in the observed output. Within input region 70 to 80, 5 reference outputs are sampled per input, again from a normal density with a mean at the sine of the input, simulating multiple noisy observations per input. In the following sections, several variations of GPs are fitted toward this simulated training data, and the hypothesised means and standard deviations of the posteriors are plotted in green. The ideal expression of uncertainty is for the model to predict low uncertainty at input regions where the reference outputs are noise-free, high distributional uncertainty at input regions without training data, and high data uncertainty at input regions where the reference outputs are noisy.

6. DATA UNCERTAINTY IN INFERENCE

A GP computes interpretable distributional uncertainty by hypothesising highly uncertain outputs for query inputs that reside far from the training inputs. It also reduces model uncertainty through a marginalisation over functions. This section discusses how data uncertainty can be modelled in a GP.

6.1. Exact Gaussian process

The posteriors of both the standard (9) and joint (21) GPs have variances that are independent of the training set reference outputs. The posterior variance is thus not influenced by whether different reference outputs are close together or not, for the same or nearby inputs. This suggests that the GP may not be suited to compute data uncertainty [5]. This is illustrated in Fig. 1a and 1b, showing similar hypothesised output standard deviations, regardless of whether the reference outputs at those input regions are noise-free or noisy.

6.2. Inducing variables approximation

This paper proposes that a form of variational approximation to the GP may be able to better model data uncertainty. Before describing how a variational GP may model data uncertainty, the more common motivations for the variational approximation are first explained. The variational GP is originally motivated by its reduction of the computational cost and its ability to handle non-Gaussian output densities, rather than for data uncertainty. A GP can be computationally expensive to use, because of the need to perform matrix inversions in the gradient computation of (5), and during inference in (7) and (8). This cost scales cubically with the training set size. Furthermore, the dependence between the outputs for different training data points in (5) encumbers expressing this criterion as an expectation over data points, thereby hindering application of mini-batch optimisation to reduce this computational cost.

Sparsifying the training data helps to alleviate these issues [20]. That is, to represent the training data by a smaller number of inducing points. The matrix dimensions will then scale with the number of inducing points instead of the training set size, thereby reducing the computational cost. The variational approximation [21] is one such sparsification method, where the prior in (1) and posterior in (6) are treated as the same, and represented by the approximate density of

$$q\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}\right)=\int p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}},\boldsymbol{u};\mathbf{Z}\right)q\left(\boldsymbol{u}\right)d\boldsymbol{u}, \tag{22}$$

where inducing inputs, $\mathbf{Z}$, and corresponding inducing latent variables, $\boldsymbol{u}$, summarise the training data. To allow greater expressiveness, the inducing latent variables are not represented as individual points, but instead as a density over the outputs. A Gaussian density can be used, such that $q\left(\boldsymbol{u}\right)=\mathcal{N}\left(\boldsymbol{u};\boldsymbol{m},\mathbf{S}\right)$, where $\boldsymbol{m}$ and $\mathbf{S}$ represent the mean and covariance respectively. The variational parameters are $\mathbf{Z}$, $\boldsymbol{m}$, and $\mathbf{S}$. The output posterior of (9) is then approximated as

$$q\left(\widehat{y}|\widehat{\boldsymbol{x}}\right)=\int p\left(\widehat{y}\middle|\widehat{f}\right)q\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}\right)d\widehat{f} \tag{23}$$

$$=\mathcal{N}\left(\widehat{y}\middle|\boldsymbol{a}^\top\boldsymbol{m},k^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}}+\boldsymbol{a}^\top\left[\mathbf{S}-\mathbf{K}^{\mathbf{ZZ}}\right]\boldsymbol{a}+\sigma^2\right), \tag{24}$$

where $\boldsymbol{a}=\mathbf{K}^{\mathbf{ZZ}\,-1}\boldsymbol{k}^{\mathbf{Z}\widehat{\boldsymbol{x}}}$.

The inducing latent variables, $\boldsymbol{u}$, represent noise-free inducing outputs. The diagonal of $\mathbf{S}$ represents the variance and uncertainty of the value that each inducing latent variable should take. The approximate output posterior in (24) contains $\mathbf{S}$ in its covariance. Thus, the uncertainty about what value each inducing latent variable should take does influence the posterior uncertainty. This differs from the posteriors in (9) and (21), where the posterior variance is independent of the training set reference outputs.

6.3. Marginalise over reference outputs

When multiple reference output samples are available, these multiple outputs can be used to form a reference output density, $p^{\text{ref}}\left(\boldsymbol{y}\right)$. Taking inspiration from (23), this paper proposes that the posterior can instead be expressed as a marginalisation over all possible training set reference outputs,

$$p\left(\widehat{y}|\widehat{\boldsymbol{x}}\right)=\int p\left(\widehat{y}|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X}\right)p^{\text{ref}}\left(\boldsymbol{y}\right)d\boldsymbol{y} \tag{25}$$

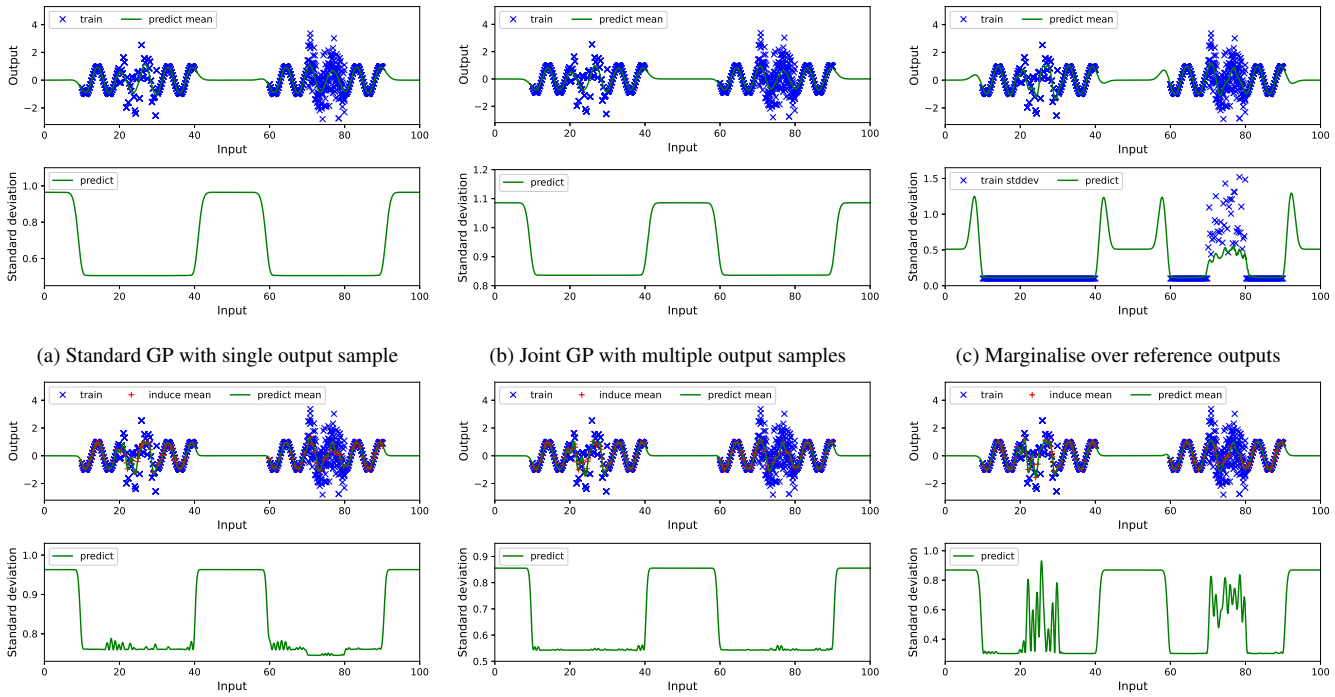
By using an approximation of independent Gaussians,

$$p^{\text{ref}}\left(\boldsymbol{y}\right)\approx\prod_i\mathcal{N}\left(y_i;\overline{\mu}_i,\overline{\eta}_i^2\right), \tag{26}$$

where the unbiased standard deviation of the multiple outputs is $\overline{\eta}_i=\sqrt{\frac{R_i}{R_i-1}}\overline{\eta}_i'$, (25) becomes

$$p(\widehat{y}|\widehat{\boldsymbol{x}})=\mathcal{N}\Big(\widehat{y}\Big|\boldsymbol{b}^\top\overline{\boldsymbol{\mu}},k^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}}+\boldsymbol{b}^\top\Big[\mathbb{D}(\overline{\boldsymbol{\eta}})-\mathbf{K}^{\mathbf{XX}}-\sigma^2\mathbf{I}\Big]\boldsymbol{b}+\sigma^2\Big), \tag{27}$$

where $\boldsymbol{b}=[\mathbf{K}^{\mathbf{XX}}+\sigma^2\mathbf{I}]^{-1}\boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}}$. The posterior variance here is dependent on the multiple reference output samples, analogously to (24). The standard deviation of the reference outputs affects the posterior variance, thereby allowing data uncertainty evident in the training data to influence the data uncertainty expressed at run-time. This is illustrated



(a) Standard GP with single output sample

(b) Joint GP with multiple output samples

(c) Marginalise over reference outputs

(d) Variational GP with multiple output samples

(e) Normalise (d) by number of output samples

(f) Interpolate proposed KL divergence criterion

Fig. 1: Toy example to demonstrate the ability of the GP to model uncertainty

in Fig. 1c without hyper-parameter tuning, showing a larger hypothesised output standard deviation at input region 70 to 80, where there are multiple noisy output reference samples per input.

Although this approach allows explainable data uncertainty in the posterior, it is not obvious what training criterion to use to optimise the GP hyper-parameters. The reference outputs already appear in the reference density on the right side of (25). The absence of hyper-parameter optimisation in Fig. 1c yields poor calibration of the prediction. Furthermore, Fig. 1c suggests that data uncertainty is only captured where there are multiple reference output samples per input, and not where there is only a single noisy reference output. This approach follows a Bayesian philosophy, by marginalising over the uncertainty about what the reference output should be. The joint marginal log-likelihood optimisation in (16) is instead a frequentist approach, by maximising the likelihood of what has been observed.

7. DATA UNCERTAINTY IN TRAINING

The previous section has discussed that either a variational approximation or Bayesian approach can allow the GP posterior to model data uncertainty. This section discusses how the GP hyper and variational parameters can be optimised to better express data uncertainty.

7.1. Marginal likelihood

By extending the marginal log-likelihood in (5) to the joint marginal log-likelihood in (16), it is hoped that the hyper-parameters can be tuned with knowledge of the multiple reference output samples, and thereby learn from the data uncertainty exhibited in the training data. However, (15) shows that by maximising $\mathcal{F}_{\text{FML}}$, it is only the scalar noise hyper-parameter, σ, in $\boldsymbol{\omega}$ that learns to fit the uncertainty expressed in the reference standard deviation, $\overline{\boldsymbol{\eta}}'$. This scalar hyper-parameter may not suffice to learn an output data uncertainty that varies across regions of the input space. Fig. 1a and 1b show that these approaches do not adequately express data uncertainty.

7.2. Variational lower-bound

The hyper and variational parameters of the variational GP can be optimised by maximising the Evidence Lower BOund (ELBO) [22],

$$\mathcal{F}_{\text{ELBO}} = \left[\sum_i \int q(f_i|\boldsymbol{x}_i) \log p(y_i|f_i) df_i\right] - \text{KL}\{q(\boldsymbol{u}) \| p(\boldsymbol{u}; \mathbf{Z})\}, \quad (28)$$

where the Kullback-Leibler (KL) divergence between the variational inducing latent variable density and the inducing GP prior is

$$\text{KL}\left\{q\left(\boldsymbol{u}\right) \| p\left(\boldsymbol{u}; \mathbf{Z}\right)\right\} = \int q\left(\boldsymbol{u}\right) \log \frac{q\left(\boldsymbol{u}\right)}{p\left(\boldsymbol{u}; \mathbf{Z}\right)} d\boldsymbol{u} \quad (29)$$

$$p\left(\boldsymbol{u}; \mathbf{Z}\right) = \mathcal{N}\left(\boldsymbol{u}; \mathbf{0}, \mathbf{K}^{\mathbf{ZZ}}\right). \quad (30)$$

This is a lower bound to the marginal log-likelihood, $\mathcal{F}_{\text{ML}} \geq \mathcal{F}_{\text{ELBO}}$. As discussed in section 6.2, a motivation for the variational approximation is to reduce the computational cost of training. The criterion in (28) can be expressed as a sum over data points. This allows a mini-batch approximation of the gradient to be used. Moreover, it is simple to substitute in non-Gaussian output densities into (28).

This paper proposes that this variational criterion can be extended to allow multiple reference output samples, simply by changing the output density to a multiple output form, as

$$\mathcal{F}_{\text{JELBO}} = \left[\sum_i \int q(f_i|\boldsymbol{x}_i) \log p(\overline{\boldsymbol{y}}_i|f_i) df_i\right] - \text{KL}\{q(\boldsymbol{u}) \| p(\boldsymbol{u}; \mathbf{Z})\}. \quad (31)$$

This Joint ELBO (JELBO) can be shown to be a lower bound to the joint marginal log-likelihood, $\mathcal{F}_{\text{JML}} \geq \mathcal{F}_{\text{JELBO}}$. Training using this criterion is depicted in Fig. 1d. The figure suggests that the criterion may place too much emphasis on data points that have multiple reference outputs, causing the standard deviation there to shrink, which opposes the intended effect. This overemphasis can be mitigated by normalising by the number of reference output samples,

$$\mathcal{F}'_{\text{JELBO}} = \left[\sum_i \frac{1}{R_i} \int q(f_i|\boldsymbol{x}_i) \log p(\overline{\boldsymbol{y}}_i|f_i) df_i\right] - \text{KL}\{q(\boldsymbol{u})\|p(\boldsymbol{u};\mathbf{Z})\}. \tag{32}$$

Fig. 1e shows that normalisation removes the unintended reduction of the standard deviation.

However, Fig. 1d and 1e suggest that $\mathcal{F}_{\text{JELBO}}$ and $\mathcal{F}'_{\text{JELBO}}$ may not encourage the GP to learn to be more uncertain where the training data shows more output noise. When the output density of (11) is substituted into (32), the terms that are dependent on the output are

$$\mathcal{F}'_{\text{JELBO}} \propto \sum_i \frac{1}{R_i}\left[\int q(f_i|\boldsymbol{x}_i) \log \mathcal{N}(\overline{\mu}_i; f_i, \omega_i) df_i + \log \mathcal{N}\left(\overline{\eta}'_i; 0, \omega_i\right)\right]. \tag{33}$$

Similarly to $\mathcal{F}_{\text{JML}}$, only the scalar σ in ω_i directly learns to fit the reference uncertainty expressed in $\overline{\eta}'_i$ when maximising $\mathcal{F}'_{\text{JELBO}}$.

7.3. Train posterior to emulate reference density function

This paper proposes to encourage the GP hyper and variational parameters to better learn about the data uncertainty in the training data, by explicitly training the approximate posterior toward the reference density function, through a KL divergence minimisation,

$$\mathcal{F}_{\text{KL}} = -\sum_i \int p^{\text{ref}}(y_i) \log q(y_i|\boldsymbol{x}_i)\, dy_i. \tag{34}$$

The approximate posterior in (24) may capture training set data uncertainty in $\mathbf{S}$. The per-data point reference density function can be represented as either a mixture model of delta functions centred at each of the reference output samples, or a Gaussian approximation $p^{\text{ref}}(y_i) = \mathcal{N}\left(y_i; \overline{\mu}_i, \overline{\eta}_i^2\right)$, both are mathematically equivalent when substituted into (34). This criterion aims to train the posterior to emulate the uncertainty expressed in the reference density function.

However, this criterion ignores any joint dependence that the GP enforces between the outputs across different data points, and is thus somewhat mismatched with the GP formulation. Ignoring the dependence allows avoidance of the issue of determining what it may mean for multiple output samples across multiple data points to be jointly dependent. Also, (34) implies that if multiple output samples are to be sampled from the same input, then each instance would first sample a separate instance of the latent variable. This differs from (14), which instead enforces that any latent variables sampled from the same input must be perfectly correlated. Because of these mismatches, the criterion is not used on its own, but is interpolated together with the variational lower bound,

$$\mathcal{F} = \mathcal{F}'_{\text{JELBO}} - \lambda \mathcal{F}_{\text{KL}}, \tag{35}$$

where $\lambda \geq 0$ is the interpolation weight.

A variational GP trained toward (35) with $\lambda = 5$ is illustrated in Fig. 1f. This GP is able to hypothesise a larger output standard deviation at input regions where there are multiple noisy reference output samples, and surprisingly, also at input regions where there is only a single noisy output reference. This suggests that the proposed approach allows the GP to learn to express data uncertainty.

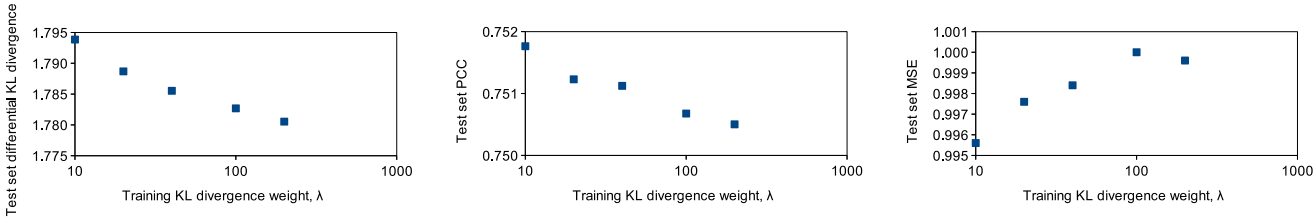
In a dataset, uncertainty about the reference output can be expressed by obtaining output samples from multiple raters. This provides a convenient reference density function to train toward and to evaluate against. However, employing multiple raters can be expensive. Fig. 1f shows that the proposed method allows the GP to learn from a limited number of data points that have multiple reference output samples, to also hypothesise reasonable output uncertainty at input regions where the training data only has single reference outputs. Thus, it may not be necessary to expend resources to collect multiple output samples for all training data points, in order for the model to learn to hypothesise reasonable data uncertainty.

8. EXPERIMENT ON SPOKEN LANGUAGE ASSESSMENT

In addition to the toy experiment in Fig. 1, the proposal is also evaluated on the speechocean762 [17] SLA dataset. This comprises 2500 English read sentences from 125 disjoint native-Mandarin speakers, in each of the training and test sets. At the sentence level, each sentence is annotated with pronunciation accuracy, fluency, prosody, sentence completion, and a holistic score. These are provided by 5 raters per sentence, as an integer between 0 and 10, where 10 is best. A scalar reference was computed as the mean of the multiple reference output samples, and compared against in the Pearson's Correlation Coefficient (PCC) and Mean Squared Error (MSE) evaluation metrics. This differs from [17], which instead uses the median. A scalar hypothesis was inferred from the GP posterior as the equivalent mean/mode/median. The PCC and MSE assess how well the hypothesised score matches the scalar reference score. However, this paper is more interested in the match between the hypothesised posterior and the reference density function, thereby assessing whether the model and group of raters express similar uncertainties about the score. This was measured by computing the unnormalised differential KL divergence of (34). Statistical significance between a pair of KL divergences was measured using a two-tailed paired t-test.

The audio and text were first force-aligned using a hybrid speech recognition model, trained on the 960 hours Librispeech data [23] toward the cross-entropy criterion using Kaldi [24], following [17]. From the alignment, Goodness Of Pronunciation (GOP) [25], Log Phone Posterior (LPP) [26], Log Posterior Ratio (LPR) [26], tempo [27], and Pitch [28] were extracted for each phone. Phone embeddings were extracted using a continuous skip-gram model [29] with a 32-node recurrent NN [30] hidden layer, trained on the non-silence training set phone sequences. These features were concatenated to form one feature vector per phone. The Wav2Vec (W2V) 2.0 XLSR-53 model [31] was also used to obtain acoustic embeddings. A learned weighted sum combined the embeddings from all transformer layers into one embedding per frame. Then, mean pooling computed one embedding per phone. When used, these embeddings would be concatenated with the previously described features.

The feature extractors and GP SLA model are illustrated in Fig. 3. A sequence of features, with one vector per phone in the sentence, was used as input to a sentence-level feature extractor. Two variations of sentence feature extractors were used, to assess the generalisability of the proposed approach. The first variation did not concatenate the W2V embeddings. The feature sequence was parsed through a Bidirectional Long Short-Term Memory (BLSTM) layer [32] with 32 nodes per direction, followed by mean pooling across the sentence to obtain one embedding per sentence. Dropout [33] with 60% omission was used. This resembles the setup in [34]. The second variation concatenated the W2V embeddings. The feature sequence was parsed through 3 8-head 24-dimensional transformer layers [35], followed by attention pooling across the sen-



(a) Differential KL divergence vs λ (b) PCC vs λ (c) MSE vs λ

Fig. 2: Interpolation of $\mathcal{F}'_{\text{JELBO}}$ and $\mathcal{F}_{\text{KL}}$ with various interpolation weights, λ

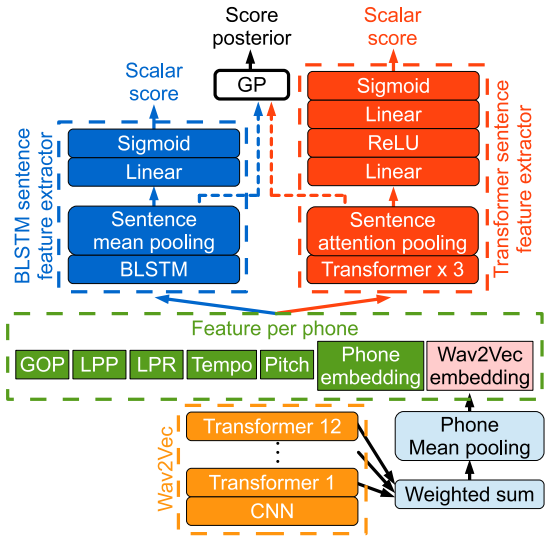


Fig. 3: GP SLA model and sentence-level feature extraction

tence. Dropout with 10% omission was used. This is inspired by [36]. The embeddings after sentence pooling were used as inputs to the GP. The feature extractors were not trained together with the GP to avoid potential overfitting [37]. Instead, the output layers shown in Fig. 3 were used to compute scalar scores from the sentence-level embeddings, which were then trained using an MSE criterion toward the scalar reference scores. The parameters were optimised using gradient descent with an AdamW [38] optimiser and early stopping on a 10% held-out validation set. The W2V parameters were frozen to avoid overfitting. Previous GP works have used handcrafted sentence-level features [18] or designed kernels to operate on sequences [39]. The GP was implemented using GPyTorch [40].

The experiment evaluates the ability of the GP to model the uncertainty expressed within the multiple reference output samples. Two setups were used, to show generalisability. The first computed the sentence-level pronunciation accuracy using the transformer sentence feature extractor with W2V embeddings concatenated at the input. The second computed the sentence-level fluency using the BLSTM sentence feature extractor without W2V embeddings.

A standard GP was trained toward $\mathcal{F}_{\text{ML}}$ with a scalar reference. A joint GP was also trained toward $\mathcal{F}_{\text{JML}}$ and inferred from using the multiple reference outputs. The results in Table 1 show that the differential KL divergence improves when training and inferring with knowledge of the multiple reference outputs. However, as is described in section 7.1 and Fig. 1b , the joint GP may not have the capacity to fit a detailed structure of the reference output uncertainty.

Table 1: Train variational GP posterior toward reference density

Model	Training	KL↓	PCC↑	MSE↓
Pronunciation accuracy transformer with W2V embeddings				
Standard GP	$\mathcal{F}_{\text{ML}}$	2.731	0.761	0.958
Joint GP	$\mathcal{F}_{\text{JML}}$	1.788	0.756	0.972
Variational joint GP	$\mathcal{F}'_{\text{JELBO}}$	1.802	0.756	0.981
	$+\mathcal{F}_{\text{KL}}$	1.783	0.751	1.000
Fluency BLSTM without W2V embeddings				
Standard GP	$\mathcal{F}_{\text{ML}}$	10.476	0.770	0.754
Joint GP	$\mathcal{F}_{\text{JML}}$	1.760	0.780	0.720
Variational joint GP	$\mathcal{F}'_{\text{JELBO}}$	1.769	0.778	0.731
	$+\mathcal{F}_{\text{KL}}$	1.744	0.779	0.732

To address this limitation, section 7.3 proposes to use a variational approximation to the GP and interpolate a KL divergence criterion during training, to encourage the approximate posterior to emulate the reference density function. 1000 inducing points were used. The results suggest that transitioning from a joint GP to a variational GP with $\mathcal{F}'_{\text{JELBO}}$ may slightly degrade the differential KL divergence, possibly due to the approximation in the model. There are an equal number of output reference samples for all datapoints, making $\mathcal{F}_{\text{JELBO}}$ and $\mathcal{F}'_{\text{JELBO}}$ proportional to each other. Using the $\mathcal{F}'_{\text{JELBO}}$ variational GP as an initialisation, the GP hyper and variational parameters were then fine-tuned toward an interpolation of $\mathcal{F}'_{\text{JELBO}}$ and the proposed $\mathcal{F}_{\text{KL}}$. Fig. 2 show the performance for various interpolation weights, using the pronunciation accuracy transformer setup. Increasing the weight of $\mathcal{F}_{\text{KL}}$ consistently improves the test set differential KL divergence, thereby suggesting a better modelling of uncertainty. However, this simultaneously degrades the PCC and MSE. Table 1 shows the performance using an interpolation weight of 100. This yields a differential KL divergence improvement over $\mathcal{F}_{\text{JML}}$ training, with significances of $\rho_{\text{KL}} < 0.0001$ for both the pronunciation accuracy transformer and fluency BLSTM setups. All GPs exhibit comparable performances in the PCC and MSE metrics, suggesting reasonable predictive performance of scalar scores.

9. CONCLUSION

This paper aims to improve the ability of a GP to express data uncertainty. The variational GP approximation has the capacity to capture data uncertainty in the training data, but standard training criteria do not encourage this to be learned. A KL divergence criterion is proposed to encourage the approximate posterior to better emulate the reference density function. The experiments show that training a variational GP with this KL divergence criterion improves the prediction of the uncertainty exhibited between different human raters.

10. REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, MIT Press, 2006.

[2] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *NeurIPS*, Montréal, Canada, Dec 2018, pp. 7047–7058.

[3] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, Sydney, Australia, Aug 2017, pp. 1321–1330.

[4] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *NIPS*, Long Beach, USA, Dec 2017, pp. 5574–5584.

[5] B. Xu, R. Kuplicki, S. Sen, and M. P. Paulus, "The pitfalls of using Gaussian process regression for normative modeling," *PLoS ONE*, vol. 16, no. 9, Sep 2021.

[6] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *ICML*, Bonn, Germany, Aug 2005, pp. 1012–1019.

[7] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *NIPS*, Vancouver, Canada, Dec 2007, pp. 153–160.

[8] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997.

[9] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, pp. 335–359, Nov 2008.

[10] C. Lea, V. Mitra, A. Joshi, S. Kajarekar, and J. P. Bigham, "SEP-28K: A dataset for stuttering event detection from podcasts with people who stutter," in *ICASSP*, Toronto, Canada, Jun 2021, pp. 6798–6802.

[11] J. Han, Z. Zhang, M. Schmitt, M. Pantic, and B. Schuller, "From hard to soft: towards more human-like emotion recognition by modelling the perception uncertainty," in *ACM international conference on Multimedia*, Mountain View, USA, Oct 2017, pp. 890–897.

[12] J. Li, R. Zhao, J-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria," in *Interspeech*, Singapore, Sep 2014, pp. 1910–1914.

[13] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Deep Learning and Representation Learning Workshop, NIPS*, Montréal, Canada, Dec 2014.

[14] G. Rasch, *Probabilistic models for some intelligence and attainment tests*, University of Chicago Press, 1960.

[15] A. Loukina, N. Madnani, A. Cahill, L. Yao, M. S. Johnson, B. Riordan, and D. F. McCaffrey, "Using PRMSE to evaluate automated scoring systems in the presence of label noise," in *Workshop on Innovative Use of NLP for Building Educational Applications*, Seattle, USA, Jul 2020, pp. 18–29.

[16] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *ICASSP*, Munich, Germany, Apr 1997, pp. 887–890.

[17] J. Zhang, Z. Zhang, Y. Wang, Z. Yan, Q. Song, Y. Huang, K. Li, D. Povey, and Y. Wang, "speechocean762: an open-source non-native English speech corpus for pronunciation assessment," in *Interspeech*, Brno, Czechia, Aug 2021, pp. 3710–3714.

[18] R. C. van Dalen, K. M. Knill, and M. J. F. Gales, "Automatically grading learners' English using a Gaussian process," in *SLaTE*, Leipzig, Germany, Sep 2015, pp. 7–12.

[19] J. H. M. Wong, H. Zhang, and N. F. Chen, "Multiple output samples per input in a single-output Gaussian process," in *Symposium for Celebrating 40 Years of Bayesian Learning in Speech and Language Processing and Beyond, ASRU (in review)*, Taipei, Taiwan, Dec 2023, arXiv preprint arXiv:2306.02719.

[20] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *JMLR*, vol. 6, no. 65, pp. 1939–1959, Dec 2005.

[21] M. K. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *AISTATS*, Cleanwater Beach, USA, Apr 2009, pp. 567–574.

[22] J. Hensman, A. G. de G. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," in *AISTATS*, San Diego, USA, May 2015, pp. 351–360.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*, Brisbane, Australia, Apr 2015, pp. 5206–5210.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *ASRU*, Hawaii, USA, Dec 2011.

[25] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2-3, pp. 95–108, Feb 2000.

[26] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166, Mar 2015.

[27] H. Zhang, K. Shi, and N. F. Chen, "Multilingual speech evaluation: case studies on English, Malay and Tamil," in *Interspeech*, Brno, Czechia, Aug 2021, pp. 4443–4447.

[28] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *ICASSP*, Florence, Italy, May 2014, pp. 2494–2498.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, Scottsdale, USA, May 2013.

[30] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, Apr 1990.

[31] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *Interspeech*, Brno, Czechia, Aug 2021, pp. 2426–2430.

[32] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *IJCNN*, Montreal, Canada, Jul 2005, pp. 2047–2052.

[33] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, Jun 2014.

[34] J. H. M. Wong, H. Zhang, and N. F. Chen, "Distilling knowledge from Gaussian process teacher to neural network student," in *Interspeech*, Dublin, Ireland, Aug 2023, pp. 426–430.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and L. Kaiser, "Attention is all you need," in *NIPS*, Long Beach, USA, Dec 2017, pp. 5998–6008.

[36] F.-A. Chao, T.-H. Lo, T.-I. Wu, Y.-T. Sung, and B. Chen, "3M: an effective multi-view, multi-granularity, and multi-aspect modeling approach to English pronunciation assessment," in *APSIPA*, Chiang Mai, Thailand, Nov 2022, pp. 575–582.

[37] S. W. Ober, C. E. Rasmussen, and M. van der Wilk, "The promises and pitfalls of deep kernel learning," in *UAI*, Jul 2021, pp. 1206–1216.

[38] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, New Orleans, USA, May 2019.

[39] H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," in *NIPS*, Denver, USA, Nov 2000, pp. 563–569.

[40] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *NeurIPS*, Montréal, Canada, Dec 2018, pp. 7587–7597.