DISTILLING DISTRIBUTIONAL UNCERTAINTY FROM A GAUSSIAN PROCESS

Jeremy H. M. Wong and Nancy F. Chen

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

ABSTRACT

A Neural Network (NN) may exhibit overconfidence about wrong hypotheses, especially for Out-Of-Domain (OOD) inputs. A Gaussian process (GP) instead has an explainable distributional uncertainty behaviour, by predicting hypotheses with greater uncertainty for query inputs further from the training data. Previous work has shown that a NN can learn to emulate the behaviour of a GP on in-domain data. This paper expands upon this, by proposing to train a NN student to emulate the GP teacher's distributional uncertainty behaviour on OOD data. This avoids the computational cost of using a GP at run-time, while improving the OOD confidence calibration of a NN. More accurate confidence calibration may better inform how the system should feedback to the user. Experiments on the SEP-28k-E stutter detection dataset suggest that distillation of such knowledge is feasible between these models.

***Index Terms*—** Distributional uncertainty, knowledge distillation, Gaussian process, deep kernel learning, stutter detection

1. INTRODUCTION

A desirable behaviour of distributional uncertainty for a model to exhibit is to express greater uncertainty about its hypotheses for query inputs further from the training data support. It is reasonable to expect that some degree of mismatch will always exist between the distributions of the training and query inputs, because of the infeasibility of collecting an infinite support of training data. A well-calibrated expression of uncertainty, when queried with such unfamiliar inputs, can provide useful information to determine the type of feedback that is given to a user. This is especially crucial in scenarios where safety is a significant concern [1]. When a model predicts an uncertain hypothesis because of an Out-Of-Domain (OOD) query, the system can be designed to seek clarification from the user, seek human intervention, or take precautionary measures, instead of only returning the potentially wrong hypothesis.

Neural Networks (NN) have been observed to be overly confident about wrong hypotheses for OOD queries [2]. As opposed to this, a Gaussian Process (GP) expresses an explainable distributional uncertainty, which is higher for query inputs that are further from the training inputs as measured by the kernel [3]. However, a GP can be expensive to use at run-time, because of the need to invert matrices that scale with the training set size or number of inducing points. The run-time computational cost of a NN does not depend on the training set size. It has previously been demonstrated that knowledge distillation [4, 5] can be used to train a NN to emulate the behaviour of a GP on in-domain data [6]. This paper proposes to investigate the feasibility of distilling the OOD distributional uncertainty behaviour from a GP teacher to a NN student. This is approached by first using the GP to compute soft labels for an OOD adaptation set, which are intended to express the GP's distributional uncertainty. A NN student is then trained toward these labels.

The approach is evaluated on a stutter detection task, which performs binary classification, aiming to detect the presence of stutter dysfluencies in speech. This is useful in educational and medical applications. In such settings, it is often difficult to obtain large quantities of diverse training data. As such, mismatches may exist between the training and run-time data. Furthermore, a well-calibrated prediction of the uncertainty of the hypothesis may be particularly important in these applications, as the system feedback may affect the quality of education or medical care that a user receives.

2. RELATED WORK

This paper uses a single model to both predict a hypothesis and express confidence in this hypothesis. Alternatively, a separate confidence estimation module can be used [7]. Ensemble methods [8] aim to reduce the influence of model uncertainty by marginalising over a diversity of multiple models, thereby yielding a purer representation of data and distributional uncertainties.

It is possible to explicitly train a NN in a supervised fashion to be able to detect inputs that are OOD, as is often done in anomaly detection methods [9]. A NN can also be trained to compute near-uniform posteriors with high entropy for OOD inputs [10]. These methods require OOD examples to be available during training.

3. GAUSSIAN PROCESS

This section describes the formulation of a GP [3], beginning with the standard exact regression approach, then expanding to a variational approximation for classification tasks. A GP computes a prior distribution over latent variables, $\boldsymbol{f}$, when given input features, $\mathbf{X}$, that is jointly Gaussian,

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

The kernel computes the Gaussian covariance, and is expressed as a short-form notation of $\mathbf{K}^{\mathbf{XX}'}$ to represent the similarity between features $\mathbf{X}$ and $\mathbf{X}'$. This paper uses a 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 i and j are data point indexes, and s and l are the scale and length hyper-parameters respectively. For a regression task with a σ hyper-parameter representing observed noise in the outputs, $\boldsymbol{y}$, the outputs are Gaussian distributed when given the latent variables, $p\left(\boldsymbol{y}|\boldsymbol{f}\right) = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{f}, \sigma^2\mathbf{I}\right)$, where $\mathbf{I}$ is the identity matrix. These outputs are assumed to be conditionally independent of the inputs when given the latent variables, resulting in a marginal likelihood of

$$p\left(\boldsymbol{y}|\mathbf{X}\right) = \int p\left(\boldsymbol{y}|\boldsymbol{f}\right) p\left(\boldsymbol{f}|\mathbf{X}\right) d\boldsymbol{f} = \mathcal{N}\left(\boldsymbol{y}; \mathbf{0}, \mathbf{K}^{\mathbf{XX}} + \sigma^2\mathbf{I}\right). \tag{3}$$

During run-time, the posterior of the latent variable for a query input, $\widehat{\boldsymbol{x}}$, is

$$p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X}\right)=\frac{p\left(\widehat{f},\boldsymbol{y}\middle|\widehat{\boldsymbol{x}},\mathbf{X}\right)}{p\left(\boldsymbol{y}|\mathbf{X}\right)}=\mathcal{N}\left(\widehat{f};\widehat{\mu},\widehat{v}\right), \tag{4}$$

where

$$\widehat{\mu}=\boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}\top}\left[\mathbf{K}^{\mathbf{XX}}+\sigma^2\mathbf{I}\right]^{-1}\boldsymbol{y} \tag{5}$$

$$\widehat{v}=k^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}}-\boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}\top}\left[\mathbf{K}^{\mathbf{XX}}+\sigma^2\mathbf{I}\right]^{-1}\boldsymbol{k}^{\mathbf{X}\widehat{\boldsymbol{x}}}. \tag{6}$$

The output posterior is then

$$p(\widehat{y}|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X})=\int p\left(\widehat{y}\middle|\widehat{f}\right)p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X}\right)d\widehat{f}=\mathcal{N}(\widehat{y};\widehat{\mu},\widehat{v}+\sigma^2). \tag{7}$$

This posterior can be seen from (6) to compute a higher variance, and thus greater output uncertainty, for query inputs that reside further from the training set inputs. This abides by the expected distributional uncertainty behaviour that a model is hoped to have.

The GP can be extended for binary classification tasks with $y\in\{0,1\}$, by squashing the infinite support of the latent variable using a normal cumulative density function, $\phi(f)=\int_{-\infty}^{f}\mathcal{N}(f';0,1)df'$, and using a Bernoulli distribution for the output [11],

$$P\left(y|f\right)=\mathcal{B}\left(y;\phi\left(f\right)\right)=\phi\left(f\right)^y\left[1-\phi\left(f\right)\right]^{1-y}. \tag{8}$$

However, substituting this output distribution into the marginal likelihood of (3) and posterior of (7) does not yield closed forms. One approach is to approximate both the latent variable prior in (1) and posterior in (4) using a variational density [12],

$$q\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}\right)=\int p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}},\boldsymbol{u};\mathbf{Z}\right)q\left(\boldsymbol{u}\right)d\boldsymbol{u}. \tag{9}$$

This introduces inducing points, comprising the inducing inputs, $\mathbf{Z}$, and the inducing latent variables, $\boldsymbol{u}$, which summarise the training data. Using a Gaussian density for $q\left(\boldsymbol{u}\right)=\mathcal{N}(\boldsymbol{u};\boldsymbol{m},\mathbf{S})$, where $\boldsymbol{m}$ and $\mathbf{S}$ are the mean and covariance respectively, yields a closed form Gaussian of the approximate latent variable posterior in (9). Using this approximation, the output posterior can be computed as

$$P(\widehat{y}|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X})\approx\int P\left(\widehat{y}\middle|\widehat{f}\right)q\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}\right)d\widehat{f} \tag{10}$$

$$=\mathcal{B}\left(\widehat{y};\phi\left(\frac{\boldsymbol{a}^\top\boldsymbol{m}}{\sqrt{1+k^{\widehat{\boldsymbol{x}}\widehat{\boldsymbol{x}}}+\boldsymbol{a}^\top\left[\mathbf{S}-\mathbf{K}^{\mathbf{ZZ}}\right]\boldsymbol{a}}}\right)\right), \tag{11}$$

where $\boldsymbol{a}=\mathbf{K}^{\mathbf{ZZ}\,-1}\boldsymbol{k}^{\mathbf{Z}\widehat{\boldsymbol{x}}}$. The variational parameters are $\mathbf{Z}$, $\boldsymbol{m}$, and $\mathbf{S}$. The hyper and variational parameters can be optimised by maximising the Evidence Lower BOund (ELBO), which is a lower bound to the marginal log-likelihood [13]. In addition to handling the non-Gaussian nature of the Bernoulli output for classification tasks, the introduction of inducing variables in the variational approximation also reduces the run-time computational cost to scale with the number of inducing points instead of the training set size and allows for mini-batch optimisation.

4. DISTILLING DISTRIBUTIONAL UNCERTAINTY

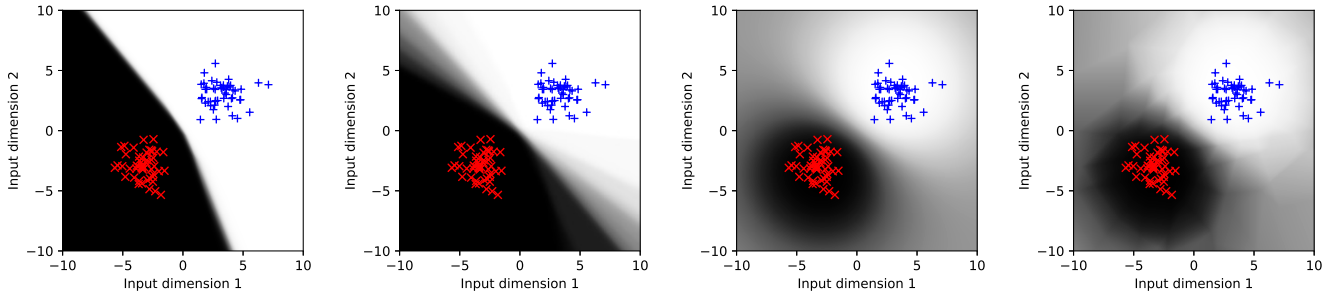
The uncertainty that a model may express can be taxonomised into aleatoric and epistemic [14]. Aleatoric uncertainty, also known as data uncertainty, is that which cannot be reduced by collecting more training data points. It may arise from observational noise, class overlaps in the input space, and subjectivity in deciding the output labels. Epistemic uncertainty is that which can be reduced by collecting more training data points. This comprises model and distributional uncertainties, the latter of which is the focus of this paper. Model uncertainty is not knowing which model topology or parameter set is the most appropriate for the task, when the criterion value for multiple non-equivalent models is computed as the same over a finite training set. Collecting more training data points from either within or outside the existing training support may resolve between these competing models. Distributional uncertainty is the expectation that outputs for query inputs outside the training data support should be hypothesised with greater uncertainty. Collecting more data points outside the existing training support expands this support, thereby reducing the distributional uncertainty within those regions.

Rather than trying to reduce distributional uncertainty, the methods discussed here instead aim to more accurately compute the existing distributional uncertainty. These are compared in a 2-dimensional toy example in Fig. 1, which comprised 50 training points in each of two classes, sampled from unit-variance Gaussians centred at [3,3] and [-3,-3] respectively. The NNs had 2 hidden layers of 10 nodes with ReLU activations, with a 2-class softmax output, and were trained with Binary Cross-Entropy (BCE). A variational GP used 50 inducing points, and was trained by maximising the ELBO. Training used the Adam optimiser [15] with a learning rate of 0.01 for 5000 iterations. The heat map in Fig. 1 shows the hypothesised posterior probability of one of the classes.

The result of using a single NN is shown in Fig. 1a. A decision boundary lies between the two classes. However, because of the finite number of training points, many possible local minima exist and many decision boundaries will have similar criterion values. Fig. 1a shows only one possible solution. This NN is highly confident of its predictions at all input regions away from the decision boundary, regardless of whether training data has been observed there or not.

A Bayesian NN [8] may better express uncertainty. That is, to marginalise over an ensemble of models, thereby reducing the prevalence of model uncertainty in the combined posterior, and yielding a purer expression of the remaining types of uncertainties. Markov Chain Monte Carlo (MCMC) methods [16, 17] can be used to sample a distribution of parameter sets that represent the likelihood of parameters when given the training data. However, using an ensemble can be computationally expensive at run-time. A single NN student can be trained toward such an ensemble, without needing to explicitly store all of the teachers concurrently [18]. It can also be difficult to tune MCMC to sufficiently explore the parameter support for large models. Furthermore, MCMC is often only used to explore the space of parameters, and not different model topologies. The combination of a simpler deep ensemble [19], formed by training from 20 different random parameter initialisations, is shown in Fig. 1b. This exhibits better data uncertainty, by showing a more gradual change between the class probabilities in the vicinity of the decision boundary. However, the ensemble is still very confident about its hypothesis at input regions far from the training data, which may suggest poor calibration. NN ensemble methods may therefore have limited ability to improve the expression of distributional uncertainty.

Fig. 1c shows the result when using a GP. As opposed to the single and ensemble NNs, the GP only hypothesises with high confidence within proximity of the training points. The hypothesised confidence attenuates away from the training points, even when far away from the decision boundary. This is a more desired expression of distributional uncertainty. The GP exhibits this behaviour by



(a) Single NN (b) Ensemble of NNs (c) GP (d) NN student of GP teacher

Fig. 1: Toy example of binary classification. The red $\times$ and blue $+$ show the training data points of the two classes, sampled from normal density functions centred at [-3,-3] and [3,3] respectively. The heat map shows the hypothesised probability of one of the classes.

comparing the similarity of the query input to each training set input or inducing input, then hypothesising an output that is highly correlated with the reference outputs for nearby training inputs. However, this requires a run-time computational cost that scales with the training set size or number of inducing points. As opposed to this, a NN instead stores information about the training data into its parameters, then parses the query input through those parameters to predict a hypothesis. The training set is not used during run-time, and thus does not influence the run-time computational cost of the NN.

This paper then asks whether it is possible for a NN to learn this distributional uncertainty behaviour from a GP, given their different operating principles, as prior work suggests that differences between the teacher and student designs may affect the learning [20]. A motivation for using a NN instead of a GP is to avoid the high run-time computational cost. A grid of 2000 adaptation inputs was sampled uniformly from -10 to 10 across both input dimensions, representing inputs both inside and outside the support of the original training data. The GP posteriors for these adaptation points were used as soft teacher labels. A NN student was trained on the adaptation points toward these labels by minimising the Kullback-Leibler (KL) divergence [4] between the NN and GP posteriors. The NN student's behaviour is shown in Fig. 1d. The NN student is able to emulate the GP, in exhibiting lower confidence further away from the training data, even at input regions away from the decision boundary.

5. EXPERIMENTS ON STUTTER DETECTION

The proposed distillation of distributional uncertainty behaviour from a GP to a NN is also evaluated on real data from the SEP-28k-E stutter detection dataset [21], which is an extension of SEP-28k [22]. The task is to classify the presence or absence of various stuttering dysfluency events in each audio clip. The *E* partitioning of the dataset comprises 15213, 6402, and 6562 clips in each of the training, dev, and test sets. Each audio clip is approximately 3 seconds long. This dataset, and in particular the *E* data partitioning, is chosen for this experiment because it has a speaker split that is conducive for an experiment that requires OOD adaptation, represented by a significant speaker mismatch. The training set comprises many audio clips, but only from 4 speakers. The dev and test sets comprise more speakers each. There is no speaker overlap between the sets. The presence of only 4 speakers in the training set suggests a domain mismatch between the train and test set speakers. 10% of the training clips were held out for validation. In these experiments, the dev set was reserved as adaptation data. Therefore, the classification decision threshold was not tuned on the dev set, and a default threshold of 0.5 was used instead when computing the F1 score. The data is labelled with the stuttering types of prolongations, blocks, sound repetitions, and word repetitions. If any of these are present, the clip is considered as containing stutter, labelled in the dataset as the absence of *NoStutteredWords*. In addition, the clips are also labelled with the presence of interjection of filler words. Speaker gender meta information is available for the training set.

Inspired by [23], the Wav2Vec 2.0 XLSR-53 model [24], referred to as W2V, was first used to extract embeddings from the audio. A learned weighted sum pooled the embeddings across all 12 transformer encoder layers. A linear projection to 256 dimensions was then used, followed by 3 transformer encoder layers with 8 attention heads. The sequence was compressed into a single vector per clip, using single-headed self-attention pooling. Multiple parallel output sub-networks were then branched from this pooled vector for multi-task training. Each output sub-network comprised a ReLU hidden layer and a 1-dimensional sigmoid layer for the NN models. The GP models followed a deep kernel learning framework [25]. In each output sub-network, the ReLU activation was projected to 32 dimensions using a linear layer. This was then used as the input to a variational classification GP with 2000 inducing points. A separate GP was used for each output task. Dropout with 10% omission was used. The W2V parameters were frozen to avoid overfitting. In the GP setup, the remaining model parameters were jointly trained with the GP hyper and variational parameters. The AdamW optimiser was used [26]. The GP was implemented using GPyTorch [27].

Table 1: Performance of NN and GP models with various training criteria, on SEP-28k-E test set, for *NoStutteredWords* classification

Model	Train	F1↑	NCE↑	AUC-ROC↑	AUC-PR↑	Entropy
NN	BCE	0.78	0.05	0.70	0.40	0.43
	FL	0.67	0.05	0.65	0.43	0.63
	CCC	0.78	-0.05	0.68	0.40	0.36
GP	ELBO	0.77	0.03	0.69	0.43	0.43

In [23], multi-task training is done with three tasks, namely classification of *NoStutteredWords* with Focal Loss (FL) [28], gender with FL, and one specific stuttering type with Concordance Correlation Coefficient (CCC) loss. The use of FL instead of BCE is intended to place more emphasis on more difficult to classify sam-

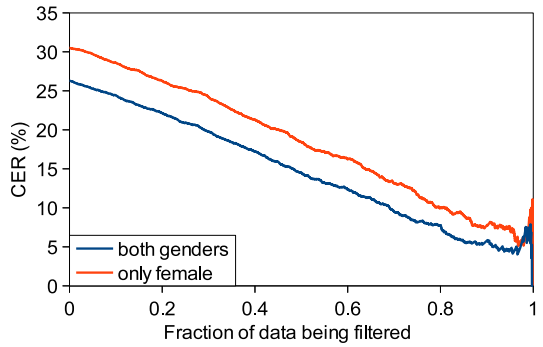


Fig. 2: Test set performance at various confidence filtering thresholds, of NNs trained on all training data or only female speakers

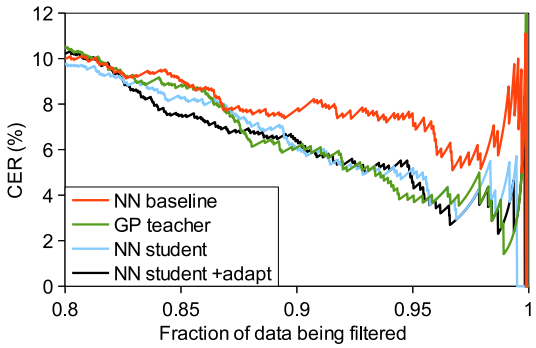


Fig. 3: Test set performance at various confidence filtering thresholds, when training a NN to emulate a GP on female-only training data, or all training and dev data in "+adapt". The NN baseline and GP teacher were trained on female-only training data

ples, which may relate to the minority class when label imbalance is significant. Work in [29] further extends the multi-task approach by training toward all stuttering types using FL. Inspired by this, the models in this paper were trained toward the classification labels of all stuttering types concurrently, together with *NoStutteredWords*, interjection, and gender, which made up 7 tasks. The experiments treated the overall presence of stutter (i.e. *NoStutteredWords*) in a clip as the main evaluated task. Table 1 compares the test set performance of NN models trained with either the BCE, FL, or CCC criteria. The default FL hyper-parameters of $\alpha = 0.25$ and $\gamma = 2$ were used. The same criterion was used for all tasks. A 7-task GP model was also trained using ELBO. Following [21, 23, 29], the classification performance was measured by the F1 score. The Normalised Cross-Entropy (NCE) [30], and Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) [31] and negative-class Precision-Recall (PR) curve [32] measure the calibration of the hypothesised confidence scores, and therefore evaluate the uncertainty modelling effectiveness. The posterior entropy, averaged over the test set was also measured. The results suggest that FL seems to degrade the F1 score, perhaps due to the lack of tuning of the FL hyper-parameters. The training set has a fairly even split between samples that are labelled with and without *NoStutteredWords*. Such a lack of imbalance may also not benefit from FL training. BCE and CCC training have comparable performances. BCE was used for NN training in the remaining experiments. The GP has a comparable F1 score with the BCE-trained NN, suggesting that the GP has a reasonable standard of performance. However, the GP does not have consistent improvements of the uncertainty calibration measures of NCE, AUC-ROC, AUC-PR, and entropy compared to the NN. These measures may be too coarse to be able to isolate effects specific to improvements in distributional uncertainty modelling.

An alternative method of visualising uncertainty calibration is presented in [33, 34]. A threshold can be placed to filter out hypotheses with confidences below it. A Classification Error Rate (CER) can then be computed only over the remaining hypotheses. This CER can be plotted against the fraction of data being filtered, as the threshold is varied. As the confidence threshold is raised, more hypotheses are filtered out of the CER computation. A well-calibrated expression of uncertainty should yield trend lines with a monotonically decreasing CER as more data is filtered out. A well-calibrated confidence should also yield as low as possible a CER at each fraction of filtered data. If a model predicts high confidence for many wrong hypotheses, this behaviour may manifest itself as a deviation from monotonicity, with an increase in the CER toward the right side of the plot, where the filtering fraction is high. The trend of the NN is shown as the dark blue line in Fig. 2. The CER increase at high filtering fractions shows that the NN can be overconfident about wrong hypotheses on the test set, which has a significant speaker mismatch with the training set. To amplify this poor confidence trend, and thereby improve the observability of follow-up trends, the speaker mismatch was exacerbated by excluding the two male speakers from the training set, and training the model on only the two remaining female speakers. The paper aims to assess the feasibility of distilling distributional uncertainty between the models, and does not seek to build the best performing system. Thus, an amplification of the experimental trends makes such assessment clearer. 10% of the female clips were held out for validation. The gender classification task was excluded in this multi-task setup, as only female training speakers were used. The trend in red shows a worsened uncertainty calibration of this female-only NN, as is intended to allow for more evident trends in the subsequent experiments.

Fig. 3 assesses the confidence of a NN student distilled from a GP teacher. The figure focuses on high data filtering thresholds, and the female-only NN performance is repeated in red. The trend for a GP trained on only the two female speakers is shown in green. The GP has a smaller increase in CER at the right side of the plot, suggesting fewer instances of high confidence for wrong hypotheses in the speaker-mismatched test set. A NN student was then trained toward the output posteriors of this GP, using only the two female training speakers, by minimising the KL divergence. Distillation was concurrently performed over six tasks, excluding gender classification, to propagate more information than just one task. This, shown in light blue, emulates the GP teacher's improved uncertainty. A NN student was also trained toward the GP posteriors on all four training speakers and the dev set OOD adaptation data. However, no further improvement of the uncertainty is seen in the black "+adapt" line for this NN student. These results suggest that a NN can effectively learn from the distributional uncertainty behaviour of a GP, even when only distilling knowledge from in-domain inputs.

6. CONCLUSION

This paper proposes to train a NN to emulate the explainable distributional uncertainty behaviour of a GP. The experiment results support the feasibility of distilling this knowledge between these types of models. A NN can therefore be used with reasonable distributional uncertainty, without the run-time computational cost of a GP.

7. REFERENCES

[1] R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, and A. Weller, "Concrete problems for autonomous vehicle safety: advantages of Bayesian deep learning," in *IJCAI*, Melbourne, Australia, Aug 2017, pp. 4745–4753.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, May 1992.

[9] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: a review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 38:1–38:38, Mar 2021.

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

[11] C. K. I. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, Dec 1998.

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

[13] 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.

[14] 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.

[15] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *ICLR*, San Diego, USA, May 2015.

[16] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *ICML*, Bellevue, USA, Jun 2011, pp. 681–688.

[17] R. M. Neal, "Bayesian learning via stochastic dynamics," in *NIPS*, Denver, USA, Nov 1992, pp. 475–482.

[18] A. Korattikara, V. Rathod, K. Murphy, and M. Welling, "Bayesian dark knowledge," in *NIPS*, Montréal, Canada, Dec 2015, pp. 3438–3446.

[19] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NIPS*, Long Beach, USA, Dec 2017, pp. 6402–6413.

[20] J. H. M. Wong, M. J. F. Gales, and Y. Wang, "Learning between different teacher and student models in ASR," in *ASRU*, Singapore, Dec 2019, pp. 93–99.

[21] S. P. Bayerl, D. Wagner, E. Nöth, T. Bocklet, and K. Riedhammer, "The influence of dataset partitioning on dysfluency detection systems," in *Text, Speech, and Dialogue*, Brno, Czech Republic, Sep 2022, pp. 423–436.

[22] 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.

[23] S. P. Bayerl, D. Wagner, E. Nöth, and K. Riedhammer, "Detecting dysfluencies in stuttering therapy using wav2vec 2.0," in *Interspeech*, Incheon, South Korea, Sep 2022, pp. 2868–2872.

[24] 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.

[25] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *AISTATS*, Cadiz, Spain, May 2016, pp. 370–378.

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

[27] 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.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, Feb 2020.

[29] S. P. Bayerl, D. Wagner, I. Baumann, F. Hönig, T. Bocklet, E. Nöth, and K. Riedhammer, "A stutter seldom comes alone - cross-corpus stuttering detection as a multi-label problem," in *Interspeech*, Dublin, Ireland, Aug 2023, pp. 1538–1542.

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

[31] A. Ragni, Q. Li, M. J. F. Gales, and Y. Wang, "Confidence estimation and deletion prediction using bidirectional recurrent neural networks," in *SLT*, Athens, Greece, Dec 2018, pp. 204–211.

[32] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *ICML*, Pittsburgh, USA, Jun 2006, pp. 233–240.

[33] D. Qiu, Q. Li, Y. He, Y. Zhang, B. Li, L. Cao, R. Prabhavalkar, D. Bhatia, W. Li, K. Hu, T. N. Sainath, and I. McGraw, "Learning word-level confidence for subword end-to-end ASR," in *ICASSP*, Toronto, Canada, Jun 2021, pp. 6393–6397.

[34] W. Liu and T. Lee, "Utterance-level neural confidence measure for end-to-end children speech recognition," in *ASRU*, Cartagena, Colombia, Dec 2021, pp. 449–456.