Uncertainty Estimation for Multi-view Data: The Power of Seeing the Whole Picture

Myong Chol Jung
Monash University
david.jung@monash.edu

He Zhao
CSIRO's Data61
he.zhao@ieee.org

Joanna Dipnall
Monash University
jo.dipnall@monash.edu

Belinda Gabbe
Monash University
belinda.gabbe@monash.edu

Lan Du*
Monash University
lan.du@monash.edu

Abstract

Uncertainty estimation is essential to make neural networks trustworthy in real-world applications. Extensive research efforts have been made to quantify and reduce predictive uncertainty. However, most existing works are designed for unimodal data, whereas multi-view uncertainty estimation has not been sufficiently investigated. Therefore, we propose a new multi-view classification framework for better uncertainty estimation and out-of-domain sample detection, where we associate each view with an uncertainty-aware classifier and combine the predictions of all the views in a principled way. The experimental results with real-world datasets demonstrate that our proposed approach is an accurate, reliable, and well-calibrated classifier, which predominantly outperforms the multi-view baselines tested in terms of expected calibration error, robustness to noise, and accuracy for the in-domain sample classification and the out-of-domain sample detection tasks[2].

1 Introduction

Reliable uncertainty estimation is critical for deploying deep learning models in a number of domains such as medical imaging diagnosis [35] or autonomous driving [8]. Even with accurate predictions, domain experts still raise questions of how trustworthy the models are [39]. For example, when a model's prediction contradicts a domain expert's opinion, the quantification of the uncertainty of the model's predictions can help determine model's reliability and justify model use.

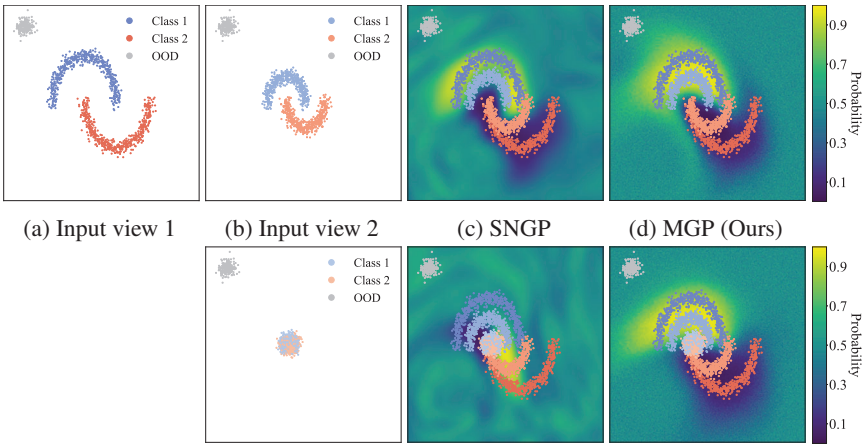
Recently, uncertainty estimation of neural networks has been an active research area, where many methods of quantifying uncertainty in predictions have been proposed [44, 9, 40, 50, 32, 24]. The majority of existing work focuses on uncertainty estimation for unimodal data. However, in many practical problems, data can exhibit in multi-views or multi-modalities. For example, LiDAR, radar, and RGB cameras can simultaneously capture complementary information about a scene [49], and computed tomography (CT) scans and x-ray images can be analyzed together to diagnose a disease [3]. Trustworthy uncertainty estimation with multi-view or multi-modalities data is important because the challenges it faces may differ from a unimodal setting (e.g., maintaining accurate predictions with one of the input views' domain shifted). Despite the success of existing work on unimodality, modelling and estimating uncertainty for multi-view data remain a less explored question [11].

*Corresponding author

[2]We provide our code at https://github.com/davidmcjung/multiview_uncertainty_estimation

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-0472



(a) Input view 1 (b) Input view 2 (c) SNGP (d) MGP (Ours)

(e) Noisy view (NV) (f) SNGP with NV (g) MGP with NV

Figure 1: Visualization on a synthetic multi-view moon dataset. Top row: the dataset has two views and two classes (e.g., blue upper circles in (a) and (b) are two views of Class 1), and an OOD class (grey); (c) and (d) are the predictive probability surfaces of SNGP and our MGP. Bottom row: A new noisy view (e) is added to the data; (f) and (g) are the predictive probability surfaces of SNGP and our MGP with the noisy view. The darker the region is (i.e., dark blue), the lower the probability of being class 1. Since SNGP is a unimodal model, input views are fused into a unimodal dataset. The difference between (c) and (f) shows SNGP cannot correctly capture the input shape in the presence of noise. MGP, however, is robust to noise, shown by minimal difference between (d) and (g).

A way of solving this problem is to fuse multi-modalities into one modality and directly apply existing unimodal methods. However, even the state-of-the-art unimodal model (e.g., SNGP [29]) can be prone to noise if one of the views in a multi-view dataset is noisy, as shown in Figure 1. Without the noisy view, unimodal models can produce accurate and confident predictions nearby the training domain. However, with the noisy view, the predictions become uncertain for samples even close to the training domain (see Figure 3). We also show that the existing multi-view classifiers (e.g., TMC [11]) have limited capacity to detect out-of-domain (OOD) samples in our experiments (see Table 4).

To this end, we propose the Multi-view Gaussian Process (MGP) that is a tailored framework providing intrinsic uncertainty estimation for classification of multi-view/modal data. Specifically, MGP consists of a dedicated Gaussian process (GP) expert for each view whose predictions are aggregated by the product of expert (PoE). In our proposed method, there is a natural way of capturing uncertainty by measuring the distance between training set and test samples in the reproducing kernel Hilbert space (RKHS). The contributions of our method can be summarized as follows:

1. We propose a new uncertainty estimation framework with GPs for multi-view data, which is an under-explored yet increasingly important problem in safety-critical applications.
2. The framework provides better uncertainty estimation through a product of expert model, providing more robustness in dealing with noise and better capacity of detecting OOD data.
3. We develop an effective variational inference algorithm to approximate multi-view posterior distributions in a principled way.
4. We conduct comprehensive and extensive experiments on both synthetic and real-world data, which show that our method achieves the state-of-the-art performance for uncertainty estimation of multi-view/modal data.

2 Muti-view Gaussian Process

Given training data $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_V\}$ where V is the number of views, each view consists of a training set of N samples $\boldsymbol{X}_v = \{\boldsymbol{x}_{v,i}\}_{i=1}^N$ and labels $\boldsymbol{y} = \{y_i\}_{i=1}^N$. In other words, the i^{th} data sample consists of V views $\{\boldsymbol{x}_{v,i}\}_{v=1}^V$ (e.g., the CUB dataset consists of images as the first view and captions as the second view) and y_i is the data sample's ground-truth label shared across the views. Without loss of generality, a multiview classification or regression problem can be formulated as

https://doi.org/10.52202/068431-0472

predicting $\boldsymbol{y}_*$ given testing samples $\{\boldsymbol{X}_{*,v}\}_{v=1}^{V}$. In this paper, we propose the Multi-view Gaussian Process (MGP), a novel framework for multi-view/modal data, where in a nutshell we first apply a GP to each view of the data and then combine in a principled way the predictions from all the GPs as a unified prediction by using the product of expert (PoE) [28].

2.1 GP for an Individual View

For each view, we consider a multiclass classification problem with C classes. We set C independent Gaussian priors over latent function $\boldsymbol{f}_v(\cdot)$ with zero-mean and $N \times N$ covariance matrix $\boldsymbol{K}_{NN}$ whose element is $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_{v,i}, \boldsymbol{x}_{v,j})$, where $k(\cdot,\cdot)$ is a kernel function. The radial basis function (RBF) which is commonly used in the GP literature [51, 33] is selected in this paper as the covariance function. It is defined as $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_v^2 \exp\left(\frac{-(\boldsymbol{x}-\boldsymbol{x}')^2}{2l_v^2}\right)$, where σ_v^2 is the signal variance and l_v is the length-scale for each GP which are parameters to be optimized.

To bypass the limitations of standard GPs [28], namely high computational cost $\mathcal{O}(N^3)$ and inconvenience of applying stochastic gradient descent (SGD), we propose to leverage the sparse variational GP (SVGP) [28, 14, 15, 43], which is detailed as follows. With SVGP, we introduce M ($M < N$) inducing points $\boldsymbol{Z}_v$ representing the training samples of view v with a smaller number of points, and the inducing variable $\boldsymbol{u}_v$ is the latent function evaluated at the inducing points (i.e., $\boldsymbol{u}_v = \boldsymbol{f}_v(\boldsymbol{Z}_v)$) where both $\boldsymbol{Z}_v$ and $\boldsymbol{u}_v$ are random variables to be optimized. Similar to the Gaussian prior set for the latent function, a joint prior can be set as:

$$\begin{bmatrix} \boldsymbol{f}_v \\ \boldsymbol{u}_v \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{NN} & \boldsymbol{K}_{NM} \\ \boldsymbol{K}_{NM}^{\mathrm{T}} & \boldsymbol{K}_{MM} \end{bmatrix}\right) \tag{1}$$

where we use $\boldsymbol{f}_v$ to indicate $\boldsymbol{f}_v(\boldsymbol{X}_v)$ for notation convenience. The use of inducing points can reduce the computational cost to $\mathcal{O}(M^3)$ [28]. We outline the likelihood of GPs in Section 2.2 and the posterior in Section 2.3.

2.2 GPs for Multi-view Data with PoE

Product of Experts (PoE) With one GP expert for each view, we propose to combine the GP experts into a unified prediction by using the PoE mechanism [17, 28, 7, 6]. Specifically, we aggregate posterior distributions of individual views by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) \propto \prod_v p(\boldsymbol{f}_v|\, \boldsymbol{y}) \tag{2}$$

For Gaussian posteriors with mean $\boldsymbol{\mu}_v$ and covariance $\boldsymbol{\Sigma}_v$, the aggregation using Equation (2) forms the unified posterior's mean and covariance expressed as:

$$\boldsymbol{\mu} = \left(\sum_v \boldsymbol{\mu}_v \boldsymbol{\Sigma}_v^{-1}\right) \boldsymbol{\Sigma}, \quad \boldsymbol{\Sigma} = \left(\sum_v \boldsymbol{\Sigma}_v^{-1}\right)^{-1} \tag{3}$$

Dirichlet-based Likelihood In order to apply Equation (3) to a multi-view problem, the latent function $\boldsymbol{f}_v$ in each view should refer to the same observable variable (i.e., $\mathcal{N}(\boldsymbol{a}|b, c)$ cannot be combined with $\mathcal{N}(\boldsymbol{c}|d, e)$ for $\boldsymbol{a} \neq \boldsymbol{c}$). However, in GP classification, the latent function is a non-observable *nuisance function* that is squashed through sigmoid or softmax function to estimate labels [51], which is not necessarily the same for every independent view. We alleviate this problem by reparameterizing the class labels to regression labels by:

$$\widetilde{\boldsymbol{y}}_i = \boldsymbol{f}_v(\boldsymbol{x}_{v,i}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \tilde{\boldsymbol{\sigma}}_i^2)$$

where $\widetilde{\boldsymbol{y}}_i$ is the transformed label and $\tilde{\boldsymbol{\sigma}}_i^2$ is the noise parameter fixed for all views. Since $\widetilde{\boldsymbol{y}}_i$ and $\tilde{\boldsymbol{\sigma}}_i^2$ are shared across the views, we ensure that $\boldsymbol{f}_v(\boldsymbol{x}_{v,i})$ refers to the same variable. By using the log-normal distribution, the Gaussian likelihood can be used in the log space as $p(\widetilde{\boldsymbol{y}}_i|\boldsymbol{f}_v) = \mathcal{N}(\boldsymbol{f}_v, \tilde{\boldsymbol{\sigma}}_i^2)$.

To transform the class labels to regression labels, we propose to adopt representing the class probability $\boldsymbol{\pi}_i = [\pi_{i,1}, \pi_{i,2}, ..., \pi_{i,C}]$ over a Dirichlet distribution with the categorical likelihood [33]:

$$\begin{aligned} p(y_i|\boldsymbol{\alpha}_i) &= \mathrm{Cat}(\boldsymbol{\pi}_i), \quad \text{where} \quad \boldsymbol{\pi}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i) \\ \pi_{i,c} &= \frac{g_{i,c}}{\sum_{j=1}^{C} g_{j,c}}, \quad \text{where} \quad g_{i,c} \sim \mathrm{Gamma}(\alpha_{i,c}, 1) \end{aligned} \tag{4}$$

https://doi.org/10.52202/068431-0472

where $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,C}]$ is the concentration parameters, the shape parameter for Gamma distribution is $\alpha_{i,c}$, and the scale parameter for Gamma distribution is $\theta = 1$. We approximate the Gamma distribution in (4) with $\tilde{g}_{i,c} \sim \text{Lognormal}(\tilde{y}_{i,c}, \tilde{\sigma}^2_{i,c})$ by moment matching:

$$\alpha_{i,c} = \exp\left(\tilde{y}_{i,c} + \tilde{\sigma}^2_{i,c}/2\right), \quad \alpha_{i,c} = \left(\exp\left(\tilde{\sigma}^2_{i,c}\right) - 1\right) \exp\left(2\tilde{y}_{i,c} + \tilde{\sigma}^2_{i,c}\right)$$

Thus, the transformed labels and the noisy parameter are expressed in terms of the concentration parameters:

$$\tilde{\sigma}^2_{i,c} = \log\left(1/\alpha_{i,c} + 1\right), \quad \tilde{y}_{i,c} = \log \alpha_{i,c} - \tilde{\sigma}^2_{i,c}/2 \tag{5}$$

where $\alpha_{i,c} = 1 + \alpha_\epsilon$ if $y_{i,c} = 1$ and $\alpha_{i,c} = \alpha_\epsilon$ if $y_{i,c} = 0$ with the one-hot label $y_{i,c}$. α_ϵ is a parameter to prevent the noise parameter from converging to infinity. See Appendix for the impacts of α_ϵ on the model performance. Compared with other transforming methods such as Platt scaling [38], the used Dirichlet likelihood compromises classification accuracy less and requires no post-hoc calibrations after training.

2.3 Training of the Proposed Framework

Given the priors from Section 2.1 and the Gaussian likelihood from Section 2.2, the goal of training our framework is to estimate a posterior distribution via variational inference (VI) [14, 15, 4]. By using Equation (2), we propose an aggregated variational distribution for all the views as:

$$q_{PoE}(\boldsymbol{f}) \propto \prod_v q(\boldsymbol{f}_v) \tag{6}$$

where $q(\boldsymbol{f}_v)$ is the variational distribution for each view that approximates the true posterior. We define $q(\boldsymbol{f}_v)$ as:

$$q(\boldsymbol{f}_v) \coloneqq \int p\left(\boldsymbol{f}_v | \boldsymbol{u}_v\right) q(\boldsymbol{u}_v)\, d\boldsymbol{u}_v. \tag{7}$$

where $p\left(\boldsymbol{f}_v | \boldsymbol{u}_v\right)$ is the conditional prior from Equation (1), and $q(\boldsymbol{u}_v)$ is the marginal variational distribution of $\mathcal{N}(\boldsymbol{m}_v, \boldsymbol{S}_v)$ with optimizable model parameters $\boldsymbol{m}_v$ and $\boldsymbol{S}_v$. The analytical solution of (7) is provided in Appendix. VI seeks to minimize the following Kullback–Leibler divergence (KL) between the true posterior and variational distributions:

$$\text{KL}\left[q_{PoE}(\boldsymbol{f}) || p(\boldsymbol{f} | \boldsymbol{X}, \widetilde{\boldsymbol{y}}_c)\right] \tag{8}$$

where $\widetilde{\boldsymbol{y}}_c = \{\tilde{y}_{i,c}\}_{i=1}^N$.

Lemma 1 (Additive Property of KL Divergence)**.** *If* $x = [x_1, \cdots, x_n] \in \mathcal{X}$, $p(x) = \prod_i^n p(x_i)$ *and* $q(x) = \prod_i^n q(x_i)$, *we have:*

$$\text{KL}\left[p(x) || q(x)\right] = \sum_i^n \text{KL}\left[p(x_i) || q(x_i)\right] \tag{9}$$

Theorem 2 (KL Divergence with PoE)**.** *With Equations* (2) *and* (6)*, we have:*

$$\text{KL}\left[q_{PoE}(\boldsymbol{f}) || p(\boldsymbol{f} | \boldsymbol{X}, \widetilde{\boldsymbol{y}}_c)\right] = \sum_v \text{KL}\left[q(\boldsymbol{f}_v) || p(\boldsymbol{f}_v | \widetilde{\boldsymbol{y}}_c)\right] \tag{10}$$

According to Theorem 2, the VI for the PoE splits to the VI of each expert/view. For the v^{th} view, the VI minimizes $\text{KL}\left[q(\boldsymbol{f}_v) || p(\boldsymbol{f}_v | \widetilde{\boldsymbol{y}}_c)\right]$, which can be turned into the maximization of the evidence lower bound (ELBO):

$$\text{ELBO}_v = \sum_{i=1}^N \mathbb{E}_{q(\boldsymbol{f}_{v,i})}\left[\log p(\tilde{y}_{i,c} | \boldsymbol{f}_{v,i})\right] - \beta \cdot \text{KL}\left[q(\boldsymbol{u}_v) || p(\boldsymbol{u}_v)\right] \tag{11}$$

where β is a parameter to control the KL term, similar to [16], which can be interpreted as a regularization term. Proofs of Equation (9)-(11) are provided in Appendix.

In order to apply SGD, we match the expectation of stochastic gradient of the expected log likelihood term to the full gradient by multiplying the number of batches to the log likelihood term in Equation (11) [15]. The overall loss for all experts is:

$$\mathcal{L} = -\sum_{v=1}^V \text{ELBO}_v \tag{12}$$

The training steps are summarized in Algorithm 1.

Algorithm 1: Learning MGP

Input: V views of training data $\boldsymbol{X} = \{\boldsymbol{X}_v\}_{v=1}^V$ where each view has N samples of $\boldsymbol{X}_v = \{\boldsymbol{x}_{v,i}\}_{i=1}^N$ and $\boldsymbol{y} = \{y_i\}_{i=1}^N$.

Transform: Reparameterize $\widetilde{\boldsymbol{y}}_c$ by (5)

for *minibatch* **do**
  for $v = 1$ **to** V **do**
    Compute $q(\boldsymbol{f}_v)$ by (7)
    Calculate ELBO_v by (11)
  end for
  Sum ELBOs by (12)
  SGD update $\{l_v, \sigma_v^2, \boldsymbol{Z}_v, \boldsymbol{m}_v, \boldsymbol{S}_v\}_{v=1}^V$
end for

Algorithm 2: Inference of MGP

Input: V views of testing data $\boldsymbol{X}_* = \{\boldsymbol{X}_{*,v}\}_{v=1}^V$

for $v = 1$ **to** V **do**
  Compute $q(\boldsymbol{f}_{*,v})$ by (13)
  Calculate $\gamma(\boldsymbol{X}_{*,v})$ by (17)
end for
Aggregate $q_{PoE}(\boldsymbol{f}_*)$ by (16)

Output: $\mathbb{E}[\pi_{i,c}]$ and $\mathbb{V}[\pi_{i,c}]$ of class probability by (14)

2.4 Inference on Test Samples

Given test samples $\boldsymbol{X}_* = \{\boldsymbol{X}_{*,v}\}_{v=1}^V$, the predictive distribution $p(\boldsymbol{f}_{*,v}|\widetilde{\boldsymbol{y}}_c)$ is estimated by the varitional distribution as:

$$p(\boldsymbol{f}_{*,v}|\widetilde{\boldsymbol{y}}_c) \approx q(\boldsymbol{f}_{*,v}) = \int p(\boldsymbol{f}_{*,v}|\boldsymbol{u}_v) q(\boldsymbol{u}_v)\, d\boldsymbol{u}_v \tag{13}$$

where $p(\boldsymbol{f}_{*,v}|\boldsymbol{u}_v)$ can be formed by the joint prior distribution similar to Equation (1) (see Appendix for a full derivation). Similar to Equation (6), we aggregate the predictive distributions to form $q_{PoE}(\boldsymbol{f}_*)$ that is sampled to approximate Gamma-distributed samples which in the end form the posterior of Dirichlet distribution as follows:

$$\begin{aligned}
\mathbb{E}[\pi_{i,c}] &= \int \frac{\exp(f_{i,c,*})}{\sum_j \exp(f_{i,j,*})} q_{PoE}(f_{i,c,*})\, d\boldsymbol{f}_* \\
\mathbb{V}[\pi_{i,c}] &= \int \left(\frac{\exp(f_{i,c,*})}{\sum_j \exp(f_{i,j,*})} - \mathbb{E}[\pi_{i,c}] \right)^2 q_{PoE}(f_{i,c,*})\, d\boldsymbol{f}_*
\end{aligned} \tag{14}$$

Equation (14) can be approximated with the Monte Carlo method. See Appendix for the impacts of the number of Monte Carlo samples on classification performance and inference time. The aggregated predictive distribution can also be weighted by each expert's predictive distribution by:

$$q_{PoE}(\boldsymbol{f}_*) \propto \prod_v \left(q(\boldsymbol{f}_{*,v})\right)^{\gamma(\boldsymbol{X}_{*,v})} \tag{15}$$

where $\gamma(\boldsymbol{X}_{*,v})$ is the weight controlling the influence of each expert to the aggregated prediction. The mean and covariance of $q_{PoE}(\boldsymbol{f}_*)$ with $\gamma(\boldsymbol{X}_{*,v})$ are:

$$\boldsymbol{\mu}_W = \left(\sum_v \boldsymbol{\mu}_v \gamma(\boldsymbol{X}_{*,v}) \boldsymbol{\Sigma}_v^{-1} \right) \boldsymbol{\Sigma}_W, \quad \boldsymbol{\Sigma}_W = \left(\sum_v \gamma(\boldsymbol{X}_{*,v}) \boldsymbol{\Sigma}_v^{-1} \right)^{-1} \tag{16}$$

In our experiments, we use negative entropy of predictive distribution:

$$\gamma(\boldsymbol{X}_{*,v}) = -H(q(\boldsymbol{f}_{*,v})) \tag{17}$$

Note that the original PoE [17] in Equation (6) is recovered if $\gamma(\boldsymbol{X}_{*,v}) = 1$. The intuition behind choosing negative entropy is that the experts with lower posterior entropy, which means the lower uncertainty, gain more contribution to the aggregated predictions. Please note that other choices of $\gamma(\boldsymbol{X}_{*,v})$ can also be applied such as the difference in entropy from prior to posterior [6] and negative predictive variance [7]. We obtain the better empirical results with negative entropy, but the choice of function is flexible. The inference steps are summarized in Algorithm 2.

https://doi.org/10.52202/068431-0472

3 Related Work

Uncertainty Estimation with GP GP has been one of the gold standards of uncertainty estimation because of GP's high sensitivity to domain shift. One of the common ways to implement GP with deep learning models is to place GP at the output layer on top of extracted features. The features are often extracted from deterministic deep neural networks [46, 29, 5], Bayesian neural networks [26], or graph data [30]. Similarly, MGP builds on these approaches and can be combined with various feature extractors. However, our method differs from all of above studies because these studies are designed for unimodal data. However, MGP is a multi-view GP. Other variants utilizing kernel learning for uncertainty estimation include deep GP [27] and RBF network [45].

Multi-view Learning Multi-view and multimodal learning aim for various downstream tasks by leveraging multiple data sources that describe the same event or object. Canonical correlation analysis (CCA) [18] holds a long history, which finds a common representation of multiple sources [1, 47]. Similarly, contrastive learning builds the common representation by forming positive pairs and negative pairs [42, 12]. Also, view-specific representations are learned to make models robust to missing input view [52]. Recently, it has been theoretically and empirically shown that vision-language models outperform unimodal models [19, 25, 21]. Other methods include gradient-blending [48] and hierarchical metric learning [53]. Despite these extensive studies in multi-view and multimodal learning, most of them are not mainly designed for uncertainty estimation.

Multi-view Uncertainty Estimation Few studies have been designed and evaluated for multi-view and multimodal uncertainty estimation. Multimodal regression with mixture of normal-inverse Gamma distributions yields promising uncertainty estimation and predictions with the real-world data [31]. However, this method is designed for regression, which differs from our method for classification. The closest study to ours is the trusted multi-view classifier (TMC) which combines evidence from different views by using Dempster's combination rule [11]. However, the Dempsterd's combination rule ignores predictions of conflicting views, which is an undesirable property especially in high-risk applications [11, 20]. In addition, our experiments show that TMC is overconfident about the OOD samples.

4 Experiments

4.1 Synthetic Dataset

To illustrate predictive behaviours of baselines and MGP, we constructed a multi-view synthetic dataset with the Scikit-learn's moon dataset [3] by scaling the data with three different scaling factors (the same dataset used in Figure 1). Each view consists of 1,000 data points formed as two half circles: upper circle (class 1) and lower circle (class 2). In each view, data points share the same relative locations with the same labels. Note that the points in the third view overlap each other, representing a noisy view.

Deep Ensemble [23] with late fusion [2] (DE(LF)), TMC, and MGP are comprised of dedicated classifiers for every view (view 1, 2, and 3 in Figure 2), and the predictions made in the views are combined as single prediction represented as multi-view in Figure 2. Since SNGP[4] is a single-view classifier, data points of all the views were concatenated as input.The results of SNGP are shown in Figure 1. For experimental details, see Appendix.

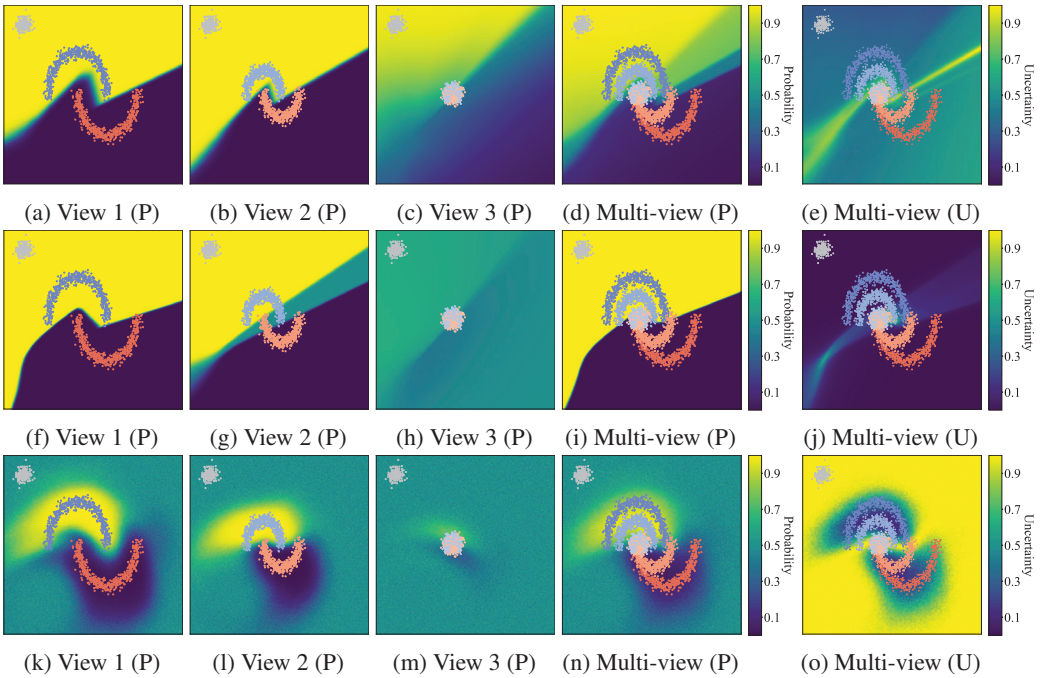
One of the benefits of having a multi-view uncertainty estimator is that the prediction accuracy of an ideal multi-view classifier remains high even if one of the input views does not provide meaningful information. This feature can be achieved by assigning lower weights to the views with high uncertainties when combining predictions. High uncertainty here refers to unconfident predictions with a uniform class probability across classes.

Figure 2 shows that the combined predictions of DE(LF) are moderately affected by the noise because the predictions are averaged across all the views. TMC and MGP, on the other hand, are not affected

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html`

[4] To provide a fair comparison, the same feature extractor without spectral normalization is used for DE(LF), TMC, MGP, and SNGP. See Appendix for results of SNGP with spectral normalization.

https://doi.org/10.52202/068431-0472



(a) View 1 (P) (b) View 2 (P) (c) View 3 (P) (d) Multi-view (P) (e) Multi-view (U)

(f) View 1 (P) (g) View 2 (P) (h) View 3 (P) (i) Multi-view (P) (j) Multi-view (U)

(k) View 1 (P) (l) View 2 (P) (m) View 3 (P) (n) Multi-view (P) (o) Multi-view (U)

Figure 2: Predictive probability surface (P) of class 1 and uncertainty estimation surface (U) from DE(LF) (a)-(e), TMC (f)-(j), and MGP (k)-(o) trained with upper circle (class 1) and lower circle (class 2) synthetic multi-view dataset. Grey data points are OOD samples. View 3 is intentionally made to be noisy. Uncertainty surfaces for individual views are plotted in Appendix.

by the noisy view (View 3). The majority of areas in the noisy view show high uncertainty (i.e., class probability close to 0.5) which have minimal impact on combining predictions because TMC and MGP are aware of uncertainty of each view. However, SNGP's prediction is heavily impacted by the noisy view as shown in Figure 1c. If SNGP were trained without any noisy view, it can properly estimate class probability (Figure 1f). This difference between Figure 1c and 1f illustrates the degrading effect of noisy view to single-view classifiers.

Although TMC is robust to noise, it produces overconfident predictions at the regions far from decision boundaries (see Figure 2f-2i). As a result, OOD samples are indistinguishable from in-domain samples (see Figure 2j). This is mainly caused by lack of distance awareness which MGP and SNGP have in common due to GP [29]. Figure 2k-2o show that MGP is well aware of the distance between the training domain and OOD by allocating high uncertainty at the OOD samples where uncertainty is calculated by the sum of variance across all classes.

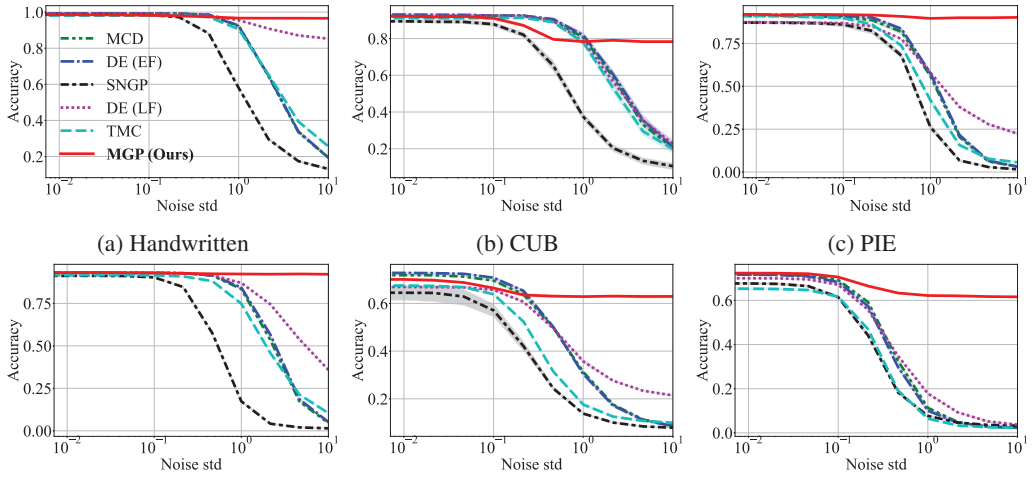
4.2 Robustness to Noise

Experimental Settings We used six multi-view datasets [11] (Handwritten, CUB, PIE, Caltech101, Scene15, and HMDB) with train-test split of 0.8:0.2. For testing robustness to noise, we added Gaussian noise of zero mean to half of the views for each dataset, following the experimental setting in [11]. In order to have the same impact of noise to all views, we normalized each view first and then added the noise because the range of raw data of each view varies significantly.[5] We increased the noise standard deviation from 0.01 to 10. To report test results invariant to selecting which views to add the noise, we ran all combinations of selecting noisy views (i.e., $\binom{V}{V/2}$ configurations) and report its average for each noise level.

Compared Methods We selected three single-view baselines and two multi-view baselines. For single-view baselines, we used early fusion (EF) technique [2] by concatenating multi-view features

[5]TMC's authors added the noise first and then normalized the noisy input. The results of this setting are reported in Appendix.

https://doi.org/10.52202/068431-0472



(a) Handwritten (b) CUB (c) PIE

(d) Caltech101 (e) Scene15 (f) HMDB

Figure 3: Domain-shift test accuracy where Gaussian noise is added to half of the views.

Table 1: In-domain test accuracy ↑

	Dataset					
Method	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
MC Dropout	**99.25±0.00**	92.33±1.09	91.32±0.62	92.95±0.29	71.75±0.25	71.68±0.36
DE (EF)	99.20±0.11	**93.16±0.70**	91.76±0.33	92.99±0.09	**72.70±0.39**	71.67±0.23
SNGP	98.85±0.22	89.50±0.75	87.06±1.23	91.24±0.46	64.68±4.03	67.65±1.03
DE (LF)	**99.25±0.00**	92.33±0.70	87.21±0.66	92.97±0.13	67.05±0.38	69.98±0.36
TMC	98.10±0.14	91.17±0.46	91.18±1.72	91.63±0.28	67.68±0.27	65.17±0.87
MGP (Ours)	98.60±0.14	92.33±0.70	**92.06±0.96**	**93.00±0.33**	70.00±0.53	**72.30±0.19**

Table 2: In-domain test ECE ↓

	Dataset					
Method	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
MC Dropout	0.009±0.000	0.069±0.017	0.299±0.005	0.017±0.003	0.181±0.003	0.388±0.004
DE (EF)	0.007±0.000	0.054±0.010	0.269±0.004	0.036±0.001	0.089±0.003	0.095±0.003
SNGP	0.023±0.004	0.200±0.010	0.852±0.012	0.442±0.004	0.111±0.063	0.227±0.010
DE (LF)	0.292±0.001	0.270±0.009	0.567±0.006	0.023±0.002	0.319±0.005	0.270±0.003
TMC	0.013±0.002	0.141±0.002	**0.072±0.011**	0.068±0.002	0.180±0.004	0.594±0.008
MGP (Ours)	**0.006±0.004**	**0.038±0.007**	0.079±0.007	**0.009±0.003**	**0.062±0.006**	**0.036±0.003**

into single feature. The selected single-view baselines are as follows: **MC Dropout** [9] with dropout rate of 0.2, **Deep Ensemble (DE)** [23] with 5 models, and **SNGP**'s GP layer [29]. The multi-view baselines are following: **Deep Ensemble (DE)** with late fusion (LF) technique [2] where one model was trained for one view and predictions of all views were combined as single prediction by averaging the predictions, and **TMC** [11]. For details of model settings, see Appendix.

Results For every metric, mean and standard deviation from five runs with different random seeds are reported. The results of multi-view classifiers are generated from the combined predictions. In Table 1 and 2, we provide test accuracy and test expected calibration error (ECE) [10] without adding noise (the definition of ECE is provided in Appendix). In terms of test accuracy, MGP outperforms DE(LF) over 4 out of 6 datasets and TMC over all the datasets. Also, the test ECE of MGP outperforms both DE(EF) and DE(LF) over all the datasets and TMC over 5 out of 6 datasets. Figure 3 shows the test accuracy with respect to the standard deviation of Gaussian noise, representing the domain-shift test accuracy. This highlights that MGP is robust to noise while the accuracy of others degrades significantly. We also report the average accuracy in Table 3.

https://doi.org/10.52202/068431-0472

Table 3: Average test accuracy with Gaussian noise (std from 0.01 to 10) added to half of the views.

Method	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
MC Dropout	82.15±0.17	76.08±0.61	64.65±0.77	73.45±0.11	48.97±0.33	42.63±0.08
DE (EF)	82.16±0.18	76.94±0.82	65.53±0.20	73.99±0.19	49.45±0.35	41.92±0.06
SNGP	72.46±0.41	61.27±1.24	56.52±0.69	56.57±0.17	38.19±1.86	37.49±0.42
DE (LF)	95.63±0.08	76.16±0.28	67.69±0.35	81.85±0.14	50.13±0.27	43.01±0.19
TMC	82.44±0.15	74.19±0.69	62.18±0.80	71.77±0.22	42.52±0.29	36.61±0.30
MGP (Ours)	**97.66±0.12**	**85.48±0.25**	**90.97±0.19**	**92.68±0.23**	**65.74±0.56**	**67.02±0.21**

Table 4: Out-of-domain detection results with CIFAR10-C

Method	Test accuracy ↑	ECE ↓	OOD AUROC ↑	
			SVHN	CIFAR100
MC Dropout	74.76±0.27	**0.013±0.002**	0.788±0.022	0.725±0.014
DE (EF)	72.95±0.13	0.154±0.048	0.769±0.008	0.721±0.014
SNGP	61.51±0.30	0.020±0.003	0.753±0.026	0.705±0.024
DE (LF)	**75.40±0.06**	0.095±0.001	0.722±0.016	0.693±0.006
TMC	72.42±0.05	0.108±0.001	0.681±0.004	0.675±0.006
MGP (Ours)	73.30±0.05	0.018±0.001	**0.803±0.007**	**0.748±0.007**

4.3 OOD Samples Detection

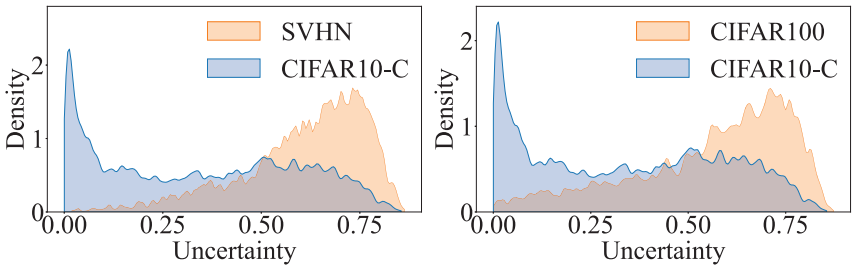
Experimental Settings Similar to experimental settings in [29], we investigated OOD detection test with CIFAR10-C [13], SVHN [36], and CIFAR100 [22]. We used CIFAR10-C as a multi-view dataset which is a corrupted version of CIFAR10 with 15 different types of corruption and 5 severity levels. The first three corruption types were selected as three multi-view inputs with severity levels of 1, 3, and 5 respectively in order to have variety of noise levels. Each view was trained with CIFAR10-C and tested with SVHN and CIFAR100 as OOD detection tests. Detecting SVHN samples is an easier task since SVHN is distinct from CIFAR10-C, and detecting CIFAR100 samples is a more difficult task since CIFAR100 looks similar to CIFAR10-C. For all methods, we used the Inception v3 [41] pre-trained with ImageNet as the feature extractor without further training it. Details of experimental settings are reported in Appendix.

Results Table 4 shows in-domain test accuracy, ECE for CIFAR10-C, and OOD results with SVHN and CIFAR100. MGP's OOD results significantly outperform the others with comparable test accuracy and ECE. Especially, MGP outperforms TMC over all the metrics. Note that the difference in ECE and OOD AUROC between TMC and MGP is considerably larger than the difference in test accuracy. The similar pattern is observed when DE(LF) and MGP are compared, where test accuracy of DE(LF) is slightly higher than MGP, but MGP outperforms DE(LF) with ECE and OOD AUROC. This highlights that although multi-view baselines can provide accurate predictions with in-domain samples, their calibration and uncertainty estimation could be limited, which aligns with [37, 34, 29, 45]. To validate that the predictive uncertainty of MGP could identify OOD samples, we also provide the predictive uncertainty with both testing sets of SVHN and CIFAR100 in Figure 4.

5 Conclusion

In this work, we have proposed a new multi-view classification framework for better uncertainty estimation and out-of-domain sample detection, where we associate each view with an uncertainty-aware GP classifier and combine the predictions of all the views with the PoE mechanism. With both the synthetic and the real-world data, we empirically demonstrated that our method is robust to domain-shift and aware of OOD samples, outperforming other baselines in ECE and OOD detection scores. Our method is not limited to a particular feature extractor and can be attached on top of existing feature extractors. A possible limitation of our work is that the weight term that balances predictive distributions is introduced in a sub-optimal way. We leave optimizing it as future work.

https://doi.org/10.52202/068431-0472



(a) SVHN (b) CIFAR100

Figure 4: Uncertatinty density of MGP with OOD testing sets: (a) SVHN and (b) CIFAR100.

Acknowledgments and Disclosure of Funding

Acknowledgments We would like to thank reviewers for their time and effort to review this paper. We sincerely appreciate the comments that enhanced and extended our work.

Funding This work was part of the Predicting fracture outcomes from clinical registry data using artificial intelligence supplemented models for evidence-informed treatment (PRAISE) study. This study is funded by the National Health and Medical Research Council of Australia Ideas Grant (NHMRC- APP2003537).

References

[1] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255, 2013.

[2] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019. doi: 10.1109/TPAMI.2018.2798607.

[3] A. Bhandary, G. A. Prabhu, V. Rajinikanth, K. P. Thanaraj, S. C. Satapathy, D. E. Robbins, C. Shasky, Y.-D. Zhang, J. M. R. Tavares, and N. S. M. Raja. Deep-learning framework to detect lung abnormality – a study with chest x-ray and lung ct scan images. *Pattern Recognition Letters*, 129:271–278, 2020. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2019.11.013.

[4] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773.

[5] J. Bradshaw, A. G. d. G. Matthews, and Z. Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

[6] Y. Cao and D. J. Fleet. Generalized product of experts for automatic and principled fusion of gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

[7] S. Cohen, R. Mbuvha, T. Marwala, and M. Deisenroth. Healing products of Gaussian process experts. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2068–2077. PMLR, 13–18 Jul 2020.

[8] D. Feng, L. Rosenbaum, and K. Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273, 2018. doi: 10.1109/ITSC.2018.8569814.

[9] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017.

[11] Z. Han, C. Zhang, H. Fu, and J. T. Zhou. Trusted multi-view classification. In *International Conference on Learning Representations*, 2021.

[12] K. Hassani and A. H. Khasahmadi. Contrastive multi-view representation learning on graphs. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4116–4126. PMLR, 13–18 Jul 2020.

[13] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[14] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA, 2013. AUAI Press.

[15] J. Hensman, A. Matthews, and Z. Ghahramani. Scalable Variational Gaussian Process Classification. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 351–360, San Diego, California, USA, 09–12 May 2015. PMLR.

[16] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[17] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, aug 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018.

[18] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. ISSN 00063444.

[19] Y. Huang, C. Du, Z. Xue, X. Chen, H. Zhao, and L. Huang. What makes multi-modal learning better than single (provably). In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10944–10956. Curran Associates, Inc., 2021.

[20] A. Jøsang. Belief fusion. In *Subjective Logic*, volume 3, chapter 12, pages 207–236. Springer, 2016.

[21] W. Kim, B. Son, and I. Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, pages 5583–5594, 2021.

[22] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[23] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[24] J. Lee, M. Humt, J. Feng, and R. Triebel. Estimating model uncertainty of neural networks in sparse information form. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5702–5713. PMLR, 13–18 Jul 2020.

 https://doi.org/10.52202/068431-0472

[25] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9694–9705. Curran Associates, Inc., 2021.

[26] Y. Li, S. Rao, A. Hassaine, R. Ramakrishnan, D. Canoy, G. Salimi-Khorshidi, M. Mamouei, T. Lukasiewicz, and K. Rahimi. Deep bayesian gaussian processes for uncertainty estimation in electronic health records. *Scientific reports*, 11(1):1–13, 2021.

[27] J. Lindinger, D. Reeb, C. Lippert, and B. Rakitsch. Beyond the mean-field: Structured deep gaussian processes improve the predictive uncertainties. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8498–8509. Curran Associates, Inc., 2020.

[28] H. Liu, Y.-S. Ong, X. Shen, and J. Cai. When gaussian process meets big data: A review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4405–4423, 2020. doi: 10.1109/TNNLS.2019.2957109.

[29] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7498–7512. Curran Associates, Inc., 2020.

[30] Z.-Y. Liu, S.-Y. Li, S. Chen, Y. Hu, and S.-J. Huang. Uncertainty aware graph gaussian process for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (04):4957–4964, Apr. 2020. doi: 10.1609/aaai.v34i04.5934.

[31] H. Ma, Z. Han, C. Zhang, H. Fu, J. T. Zhou, and Q. Hu. Trustworthy multimodal regression with mixture of normal-inverse gamma distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6881–6893. Curran Associates, Inc., 2021.

[32] A. Malinin and M. Gales. Predictive uncertainty estimation via prior networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7047–7058, Red Hook, NY, USA, 2018. Curran Associates Inc.

[33] D. Milios, R. Camoriano, P. Michiardi, L. Rosasco, and M. Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[34] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv preprint arXiv:2102.11582*, 2021.

[35] T. Nair, D. Precup, D. L. Arnold, and T. Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical Image Analysis*, 59:101557, 2020. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.101557.

[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[37] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[38] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[39] A. G. Roy, S. Conjeti, N. Navab, and C. Wachinger. Bayesian quicknat: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage*, 195:11–22, 2019. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2019.03.042.

[40] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[42] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 776–794, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58621-8.

[43] M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[44] M. Valdenegro-Toro. Deep sub-ensembles for fast uncertainty estimation in image classification. *arXiv preprint arXiv:1910.08168*, 2019.

[45] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Uncertainty estimation using a single deep deterministic neural network. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9690–9700. PMLR, 13–18 Jul 2020.

[46] J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

[47] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1083–1092, Lille, France, 07–09 Jul 2015. PMLR.

[48] W. Wang, D. Tran, and M. Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[49] Z. Wang, Y. Wu, and Q. Niu. Multi-sensor fusion in automated driving: A survey. *IEEE Access*, 8:2847–2868, 2020. doi: 10.1109/ACCESS.2019.2962554.

[50] Y. Wen, D. Tran, and J. Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

[51] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[52] C. Zhang, Z. Han, y. cui, H. Fu, J. T. Zhou, and Q. Hu. Cpm-nets: Cross partial multi-view networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[53] H. Zhang, V. M. Patel, and R. Chellappa. Hierarchical multimodal metric learning for multi-modal classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

 https://doi.org/10.52202/068431-0472

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Our contributions are supported by Section 2 and Section 4.
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix.
   (b) Did you include complete proofs of all theoretical results? See Appendix. [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4 and Appendix.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-0472