Noise Contrastive Priors for Functional Uncertainty

Danijar Hafner [*]
Google Brain

Dustin Tran
Google Brain

Timothy Lillicrap
DeepMind

Alex Irpan
Google Brain

James Davidson
Third Wave Automation

Abstract

Obtaining reliable uncertainty estimates of neural network predictions is a long standing challenge. Bayesian neural networks have been proposed as a solution, but it remains open how to specify their prior. In particular, the common practice of an independent normal prior in weight space imposes relatively weak constraints on the function posterior, allowing it to generalize in unforeseen ways on inputs outside of the training distribution. We propose noise contrastive priors (NCPs) to obtain reliable uncertainty estimates. The key idea is to train the model to output high uncertainty for data points outside of the training distribution. NCPs do so using an input prior, which adds noise to the inputs of the current mini batch, and an output prior, which is a wide distribution given these inputs. NCPs are compatible with any model that can output uncertainty estimates, are easy to scale, and yield reliable uncertainty estimates throughout training. Empirically, we show that NCPs prevent overfitting outside of the training distribution and result in uncertainty estimates that are useful for active learning. We demonstrate the scalability of our method on the flight delays data set, where we significantly improve upon previously published results.

1 INTRODUCTION

Many successful applications of neural networks (Krizhevsky et al., 2012; Sutskever et al., 2014; van den Oord et al., 2016) are in restricted settings where predictions are only made for inputs similar to the training distribution. In real-world scenarios, neural networks can face truly novel data points during inference, and in these settings it can be valuable to have good estimates of the model's uncertainty. For example, in healthcare, reliable uncertainty estimates can prevent overconfident decisions for rare or novel patient conditions (Schulam and Saria, 2015). Another application are autonomous agents that should actively explore their environment, requiring uncertainty estimates to decide what data points will be most informative.

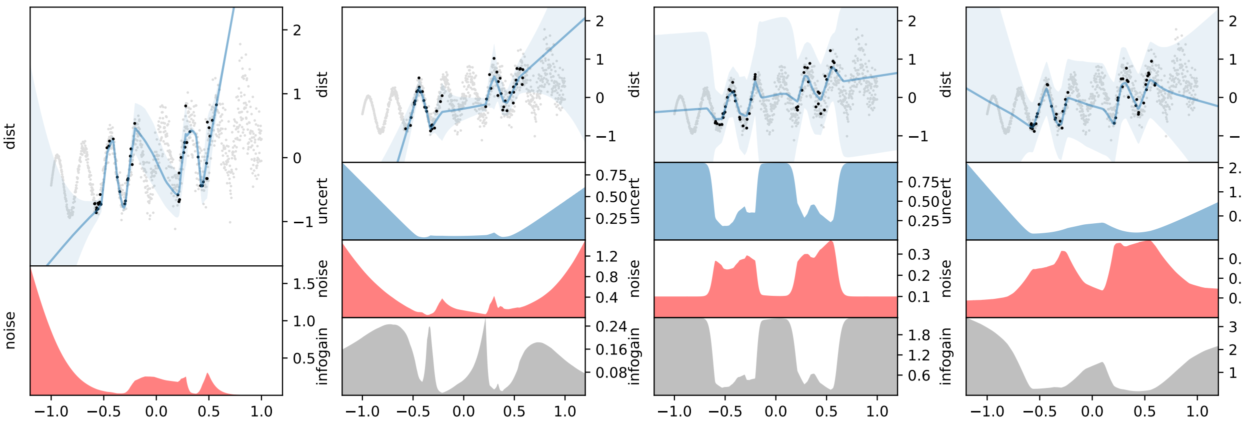
Epistemic uncertainty describes the amount of missing knowledge about the data generating function. Uncertainty can in principle be completely reduced by observing more data points at the right locations and training on them. In contrast, the data generating function may also have inherent randomness, which we call aleatoric noise. This noise can be captured by models outputting a distribution rather than a point prediction. Obtaining more data points allows the noise estimate to move closer to the true value, which is usually different from zero. For active learning, it is crucial to separate the two types of randomness: we want to acquire labels in regions of high uncertainty but low noise (Lindley et al., 1956).

Bayesian analysis provides a principled approach to modeling uncertainty in neural networks (Denker et al., 1987; MacKay, 1992b). Namely, one places a prior over the network's weights and biases. This induces a distribution over the functions that the network represents, capturing uncertainty about which function best fits the data. Specifying this prior remains an open challenge. Common practice is to use an independent normal prior in weight space, which is neither informative about the induced function class nor the data (e.g., it is sensitive to parameterization). This can cause the induced function posterior to generalize in unforeseen ways on out-of-distribution (OOD) inputs, which are inputs outside of the distribution that generated the training data.

Motivated by these challenges, we introduce noise contrastive priors (NCPs), which encourage uncertainty outside of the training distribution through a loss in data space. NCPs are compatible with any model that represents functional uncertainty as a random variable, are easy to scale, and yield reliable uncertainty estimates that show significantly improved active learning performance.

[*] Correspondence to mail@danijar.com.

Seems to be too uncertain on the training data regions though?



(a) Deterministic (b) BBB (c) ODC+NCP (d) BBB+NCP

Figure 1: Predictive distributions on a low-dimensional active learning task. The predictive distributions are visualized as mean and two standard deviations shaded. They decompose into epistemic uncertainty 🟦 and aleatoric noise 🟥. Data points are only available within two bands, and are selected using the expected information gain ▨. **(a)** A deterministic network models no uncertainty but only noise, resulting in overconfidence outside of the data distribution. **(b)** A variational Bayesian neural network with independent normal prior represents uncertainty and noise separately but is overconfident outside of the training distribution. **(c)** On the OOD classifier model, NCP prevents overconfidence. **(d)** On the Bayesian neural network, NCP produces smooth uncertainty estimates that generalize well to unseen data points. Models trained with NCP also separate uncertainty and noise well. The experimental setup is described in Section 5.1.

2 NOISE CONTRASTIVE PRIORS

Specifying priors is intuitive for small probabilistic models, where each variable often has a clear interpretation (Blei, 2014). It is less intuitive for neural networks, where the parameters serve more as adaptive basis coefficients in a nonparametric function. For example, neural network models are non-identifiable due to weight symmetries that yield the same function (Müller and Insua, 1998). This makes it difficult to express informative priors on the weights, such as expressing high uncertainty on unfamiliar examples.

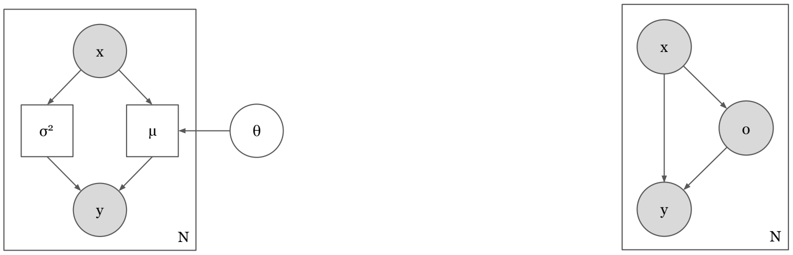
Data priors Unlike a prior in weight space, a *data prior* lets one easily express informative assumptions about input-output relationships. Here, we use the example of a prior over a labeled data set $\{x, y\}$, although the prior can also be on x and another variable in the model that represents uncertainty and has a clear interpretation. The prior takes the form $p_{\text{prior}}(x, y) = p_{\text{prior}}(x)\, p_{\text{prior}}(y \mid x)$, where $p_{\text{prior}}(x)$ denotes the *input prior* and $p_{\text{prior}}(y \mid x)$ denotes the *output prior*.

To prevent overconfident predictions, a good input prior $p_{\text{prior}}(x)$ should include OOD examples so that it acts beyond the training distribution. A good output prior $p_{\text{prior}}(y \mid x)$ should be a high-entropy distribution, representing high uncertainty about the model output given OOD inputs.

Generating OOD inputs Exactly generating OOD data is difficult. A priori, we must uniformly represent the input domain. A posteriori, we must represent the complement of the training distribution. Both distributions are typically uniform over infinite support, making them ill-defined. To estimate OOD inputs, we develop an algorithm inspired by noise contrastive estimation (Gutmann and Hyvärinen, 2010; Mnih and Kavukcuoglu, 2013), where a complement distribution is approximated using random noise.

A hypothesis of our work is that in practice it is enough to encourage high uncertainty output near the *boundary* of the training distribution, and that this effect will propagate to the entire OOD space. This hypothesis is backed up by previous work (Lee et al., 2017) as well as our experiments (see Figure 1). This means we no longer need to sample arbitrary OOD inputs. It is enough to sample OOD points that lie close to the boundary of the training distribution, and to apply our desired prior at those points.

Parameter Estimation Noise contrastive priors are data priors that are enforced on both training inputs x and inputs $\tilde{x}$ perturbed by noise. For example, in binary and categorical input domains, we can approximate



(a) Bayesian neural network

(b) Out-of-distribution classifier

Figure 2: Graphical representations of the two uncertainty-aware models we consider. Circles denote random variables, squares denote deterministic variables, shading denotes observations during training. **(a)** The Bayesian neural network captures a belief over parameters for the predictive mean, while the predictive variance is a deterministic function of the input. In practice, we only use weight uncertainty for the mean's output layer and share earlier layers between the mean and variance. **(b)** The out-of-distribution classifier model uses a binary auxiliary variable o to determine if a given input is out-of-distribution; given its value, the output mixed between a neural network prediction and a wide output prior.

OOD inputs by randomly flipping the features to different classes with a certain probability. For continuous valued inputs x, we can use additive Gaussian noise to obtain noised up inputs $\tilde{x} = x + \epsilon$. This expresses the noise contrastive prior where inputs are distributed according to the convolved distribution,

$$\begin{aligned} p_{\text{prior}}(\tilde{x}) &= \frac{1}{N}\sum_i \text{Normal}(\tilde{x} - x_i \mid \mu_x, \sigma_x^2) \\ p_{\text{prior}}(\tilde{y} \mid \tilde{x}) &= \text{Normal}(\mu_y, \sigma_y^2). \end{aligned} \tag{1}$$

The variances σ_x^2 and σ_y^2 are hyperparameters that tune how far from the boundary we sample, and how large we want the output uncertainty to be. We choose $\mu_x = 0$ to apply the prior equally in all directions from the data points. The output mean μ_y determines the default prediction of the model outside of the training distribution, for example $\mu_y = 0$. We set $\mu_y = y$ which corresponds to data augmentation (Matsuoka, 1992; An, 1996), where a model is trained to recover the true labels from perturbed inputs. This way, NCP makes the model uncertain but still encourages its prediction to generalize to OOD inputs.

For training, we minimize the loss function

$$\begin{aligned} \mathcal{L}(\theta) = &-\mathbb{E}_{p_{\text{train}}(x,y)}\big[\ln p_{\text{model}}(y \mid x)\big] \\ &+\gamma \mathbb{E}_{p_{\text{prior}}(\tilde{x})}\big[D_{\text{KL}}[p_{\text{prior}}(\tilde{y} \mid \tilde{x}) \,\|\, p_{\text{model}}(\tilde{y} \mid \tilde{x}, \theta)]\big]. \end{aligned} \tag{2}$$

The first term represents typical maximum likelihood. The second term is added by our method: it represents the analogous term on a data prior, where maximum likelihood can be derived as minimizing a KL divergence to the empirical training distribution $p_{\text{train}}(y \mid x)$ over training inputs. The hyperparameter γ sets the relative influence of the prior, allowing to trade-off between the two terms.

Interpretation as function prior The noise contrastive prior can be interpreted as inducing a function prior. This is formalized through the prior predictive distribution,

$$\begin{aligned} p(y \mid x) = \int & p_{\text{model}}(y \mid x, \theta)\, p_{\text{model}}(\theta \mid \tilde{x}, \tilde{y}) \\ & p_{\text{prior}}(\tilde{x}, \tilde{y})\, d\theta\, d\tilde{x}\, d\tilde{y}. \end{aligned} \tag{3}$$

The distribution marginalizes over network parameters θ as well as data fantasized from the data prior. The distribution $p(\theta \mid \tilde{x}, \tilde{y})$ represents the distribution of model parameters after fitting the prior data. That is, the belief over weights is shaped to make $p(y \mid x)$ highly variable. This parameter belief causes uncertain predictions outside of the training distribution, which would be difficult to express in weight space directly.

Because network weights are trained to fit the data prior, the prior acts as "pseudo-data." This is similar to classical work on conjugate priors: a $\text{Beta}(\alpha, \beta)$ prior on the probability of a Bernoulli likelihood implies a Beta posterior, and if the posterior mode is chosen as an optimal parameter setting, then the prior translates to $\alpha - 1$ successes and $\beta - 1$ failures. It is also similar to pseudo-data in sparse Gaussian processes (Quiñonero-Candela and Rasmussen, 2005).

Data priors encourage learning parameters that not only capture the training data well but also the prior data. In practice, we can combine NCP with other priors, for example the typical normal prior in weight space for Bayesian neural networks, although we did not find this necessary in our experiments.

3 VARIATIONAL INFERENCE WITH NCP

In this section, we apply a Bayesian treatment of NCP where we perform posterior inference instead of point estimation. Consider a regression task that we model as $p(y \mid x, \theta) = \text{Normal}(\mu(x), \sigma^2(x))$ with mean and variance predicted by a neural network from the inputs. This model is heteroskedastic, meaning that it can predict a different aleatoric noise amount for every point in the input space. We apply NCP to posit epistemic uncertainty on the output of the mean μ, and we infer the induced weight posterior for only the output layer (Lázaro-Gredilla and Figueiras-Vidal, 2010; Calandra et al., 2014) that predicts the mean. This results in the model

$$\theta \sim q_\phi(\theta) \qquad y \sim \text{Normal}(\mu(x, \theta), \sigma^2(x)), \tag{4}$$

where $q_\phi(\theta)$ forms an approximate posterior over weights. We do not model uncertainty about the noise estimate, as this is not required for the approximation for the Gaussian expected information gain (MacKay, 1992a) that we will use to acquire labels. The distribution of the mean induced by the weight posterior, $q(\mu(x)) = \int \mu(x, \theta) q_\phi(\theta)\, d\theta$, represents epistemic uncertainty. Note that this is different from the predictive distribution, which combines both uncertainty and noise. The loss function is

$$\begin{aligned} \mathcal{L}(\phi) = &-\mathbb{E}_{p_{\text{train}}(x,y)}\left[\mathbb{E}_{q_\phi(\theta)}[\ln p(y \mid x, \theta)]\right] \\ &+ D_{\text{KL}}[\text{Normal}(\mu_\mu, \sigma_\mu^2) \parallel q(\mu(\tilde{x}))], \end{aligned} \tag{5}$$

where $\tilde{x}$ are the perturbed inputs drawn from the input prior. Because we only use the weight belief for the linear output layer, the KL-divergence can computed analytically. In other models, it can be estimated using samples. Compared to the loss function for point estimation (Equation 2), the only difference is a posterior expectation and using the posterior predictive distribution for the KL.

Note Equation 5's relationship to the variational lower bound for typical Bayesian neural networks (Blundell et al., 2015). The expected log likelihood term is the same. Only the KL divergence differs in that it now penalizes the approximate posterior in output space rather than in weight space. This change avoids a common pathology in variational Bayesian neural network training where the variational distribution collapses to the prior; this pathology happens on a per-dimension basis as the KL decomposes into a sum of KLs for each weight dimension, making it easy for many dimensions to collapse (Bowman et al., 2015). By penalizing deviations in output space, the approximate posterior can only collapse if the entire predictive distribution is set to the output prior; this is hard to achieve as the model would pay a large cost in the data misfit (log-likelihood) term because little capacity remains to additionally fit the data.

The loss function is an (approximate) lower bound to the log-marginal likelihood. See Appendix B for its derivation via reparameterizing the original KL in weight space, resulting in the reverse KL divergence known from variational inference.

4 RELATED WORK

Priors for neural networks Classic work has investigated entropic priors (Buntine and Weigend, 1991) and hierarchical priors (MacKay, 1992b; Neal, 2012; Lampinen and Vehtari, 2001). More recently, Depeweg et al. (2018) introduce networks with latent variables in order to disentangle forms of uncertainty, and Flam-Shepherd et al. (2017) propose general-purpose weight priors based on approximating Gaussian processes. Other works have analyzed priors for compression and model selection (Ghosh and Doshi-Velez, 2017; Louizos et al., 2017). Instead of a prior in weight space (or latent inputs as in Depeweg et al. (2018)), NCPs take the functional view by imposing explicit regularities in terms of the network's inputs and outputs. This is similar in nature to Sun et al. (2019), who define a GP prior for BNNs resulting in an interesting but more complex algorithm. Malinin and Gales (2018) propose prior networks to avoid an explicit belief over parameters for classification tasks.

Input and output regularization There is classic work on adding noise to inputs for improved generalization (Matsuoka, 1992; An, 1996; Bishop, 1995). For example, denoising autoencoders (Vincent et al., 2008) encourage reconstructions given noisy encodings. Output regularization is also a classic idea from the maximum entropy principle (Jaynes, 1957), where it has motivated label smoothing (Szegedy et al., 2016) and entropy penalties (Pereyra et al., 2017). Also related is virtual adversarial training (Miyato et al., 2015), which includes examples that are close to the current input but cause a maximal change in the model output, and mixup (Zhang et al., 2018), which includes examples under the vicinity of training data. These methods are orthogonal to NCPs: they aim to improve generalization from finite data within the training distribution (interpolation), while we aim to improve uncertainty estimates outside of the training distribution (extrapolation).

Classifying out-of-distribution inputs A simple approach for neural network uncertainty is to classify whether data points belong to the data distribution, or are OOD (Hendrycks and Gimpel, 2017). Recently, Lee et al. (2017) introduce a GAN to generate OOD samples, and Liang et al. (2018) add perturbations to the input, applying an "OOD detector" to improve softmax scores on OOD samples by scaling the temperature. Extending these directions of research, we connect to Bayesian principles and focus on uncertainty estimates that are useful for active data acquisition.

5 EXPERIMENTS

To demonstrate their usefulness, we evaluate NCPs on various tasks where uncertainty estimates are desired. Our focus is on active learning for regression tasks, where only few targets are visible in the beginning, and additional targets are selected regularly based on an acquisition function. We use two data sets: a toy example and a large flights data set. We also evaluate how sensitive our method is to the choice of input noise. Finally, we show that NCP scales to large data sets by training on the full flights data set in a passive learning setting. Our implementation uses TensorFlow Probability (Dillon et al., 2017; Tran et al., 2016) and is open-sourced at `https://github.com/brain-research/ncp`. An implementation of NCP is also available in Aboleth (Aboleth Developers, 2017) and Bayesian Layers (Tran et al., 2018).

We compare four neural network models, all using leaky ReLU activations (Maas et al., 2013) and trained using Adam (Kingma and Ba, 2014). The four models are:

- **Deterministic neural network (Det)** A neural network that predicts the mean and variance of a normal distribution. The name stands for *deterministic*, as there is no weight uncertainty.
- **Bayes by Backprop (BBB)** A Bayesian neural network trained via gradient-based variational inference with a independent normal prior in weight space (Blundell et al., 2015; Kucukelbir et al., 2017). We use the same model as in Section 3 but with a KL in weight space.
- **Bayes by Backprop with noise contrastive prior (BBB+NCP)** Bayes by Backprop with NCP on the predicted mean distribution as described in Section 3.
- **Out-of-distribution classifier with noise contrastive prior (OCD+NCP)** An uncertainty classifier model described in Appendix A. It is a deterministic neural network combined with NCP which we use as a baseline alternative to Bayes by Backprop with NCP.

For active learning, we select new data points $\{x, y\}$ for which x maximizes the expected information gain under the model $q(y \mid x) = \int p(y \mid x, \theta) q(\theta)\, d\theta$,

$$\max_x \mathrm{E}_{q(y|x)}[D_{\mathrm{KL}}[q(\theta \mid x, y) \parallel q(\theta)]]. \tag{6}$$

The expected information gain is the mutual information between weights and output given input. It measures how many bits of information the new data point is expected to reveal about the optimal weights. Intuitively, the expected information gain is the largest where the model has high epistemic uncertainty but expects low aleatoric noise.

We use the form of the expected information gain for Gaussian posterior predictive distributions discussed in MacKay (1992a). Moreover, to select batches of data points, we place a softmax distribution on the information gain for all available data points and acquire labels by sampling with a temperature of $\tau = 0.5$ to get diversity. This results in the acquisition rule

$$\{x_{\text{new}}, y_{\text{new}}\} \sim p_{\text{new}}(x, y) \propto \left(1 + \frac{\mathrm{Var}[q(\mu(x))]}{\sigma^2(x)}\right)^{\frac{1}{\tau}}, \tag{7}$$

where $\sigma^2(x)$ is the estimated aleatoric noise and $q(\mu(x))$ is the epistemic uncertainty around the predicted mean projected into output space. Since our Bayesian neural networks only use a weight belief for the output layer, $q(\mu(x))$ is Gaussian and its variance can be computed in closed form. In general, it the epistemic part of the predictive variance would be estimated by sampling. In the classifier model, we use the OOD probability $p(o = 1|x)$ for this. For the deterministic neural network, we use $\mathrm{Var}[p(y \mid x)]$ as proxy since it does not output an estimate of epistemic uncertainty.

5.1 LOW-DIMENSIONAL ACTIVE LEARNING

For visualization purposes, we start with experiments on a 1-dimensional regression task that consists of a sine function with a small slope and increasing variance for higher inputs. Training data can be acquired only within two bands, and the model is evaluated on all data points that are not visible to the model. This structured split between training and testing data causes a distributional shift at test time, requiring successful models to have reliable uncertainty estimates to avoid mispredictions for OOD inputs.

For this experiment, we use two layers of 200 hidden units, a batch size of 10, and a learning rate of 3×10^{-4} for all models. NCP models use noise $\epsilon \sim \mathrm{Normal}(0, 0.5)$. We start with 10 randomly selected initial targets, and select 1 additional target every 1000 epochs. Figure 3 shows the root mean squared error (RMSE) and negative log predictive density (NLPD) throughout learning. The two baseline models severely overfit to the training distribution early on when only few data points are visible. Models with NCP outperform BBB, which in turn outperforms Det. Figure 1 visualizes the models' predictive distributions at the end of training, showing that NCP prevents overconfident generalization.

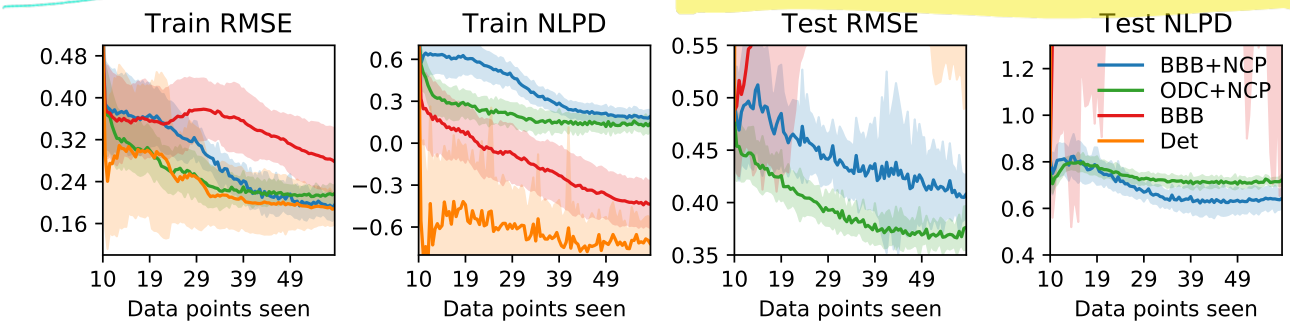


Figure 3: Active learning on the 1-dimensional regression problem, mean and standard deviation over 20 seeds. The test root mean squared error (RMSE) and negative log predictive density (NLPD) of the models trained with NCP decreases during the active learning run, while the baseline models select less informative data and overfit. The deterministic network is barely visible in the plots as it overfits quickly. Figure 1 shows the predictive distributions of the models.

5.2 ACTIVE LEARNING ON FLIGHT DELAYS

We consider the flight delay data set (Hensman et al., 2013; Deisenroth and Ng, 2015; Lakshminarayanan et al., 2016), a large scale regression benchmark with several published results. The data set has 8 input variables describing a flight, and the target is the delay of the flight in minutes. There are 700K training examples and 100K test examples. The test set has a subtle distributional shift, since the 100K data points temporally follow after the training data.

We use two layers with 50 units each, a batch size of 10, and a learning rate of 10^{-4}. For NCP models, $\epsilon \sim \mathrm{Normal}(0, 0.1)$. Starting from 10 labels, the models select a batch of 10 additional labels every 50 epochs. The 700K data points of the training data set are available for acquisition, and we evaluate performance on the typical test split. Figure 4 shows the performance for the visible data points and the test set respectively. We note that BBB and BBB+NCP show similar NLPD on the visible data points, but the NCP models generalize better to unseen data. Moreover, the Bayesian neural network with NCP achieves lower RMSE than the one without and the classifier based model achieves lower RMSE than the deterministic neural network. All uncertainty-based models outperform the deterministic neural network.

5.3 ROBUSTNESS TO NOISE PATTERNS

The choice of input noise might seem like a critical hyper parameter for NCP. In this experiment, we find that our method is robust to the choice of input noise. The experimental setup is the same as for the active learning experiment described in Section 5.2, but with uniform or normal input noise with different variance ($\sigma_x^2 \in \{0.1, 0.2, \cdots, 1.0\}$). For uniform input noise, this means noise is drawn from the interval $[-2\sigma_x, 2\sigma_x]$.

We observe that BBB+NCP is robust to the size of the input noise. NCP consistently improves RMSE for the tested noise sizes and yields the best NLPD for all noise sizes below 0.6. For our ODC baseline, we observe an intuitive trade-off: smaller input noise increases the regularization strength, leading to better NLPD but reduced RMSE. Robustness to the choice of input noise is further supported by the analogous experiment on toy data set, where above a small threshold (BBB+NCP $\sigma_x^2 \geq 0.3$ and ODC+NCP $\sigma_x^2 \geq 0.1$), NCP consistently performs well (Figure 6).

5.4 LARGE SCALE REGRESSION OF FLIGHT DELAYS

In addition to the active learning experiments, we perform a passive learning run on all 700K data points of the flights data set to explore the scalability of NCP. We use networks of 3 layers with 1000 units and a learning rate of 10^{-4}. Table 1 compares the performance of our models to previously published results. We significantly improve state of the art performance on this data set.

6 DISCUSSION

We develop *noise contrastive priors* (NCPs), a prior for neural networks in data space. NCPs encourage network weights that not only explain the training data but also capture high uncertainty on OOD inputs. We show that NCPs offer strong improvements over baselines and scale to large regression tasks.

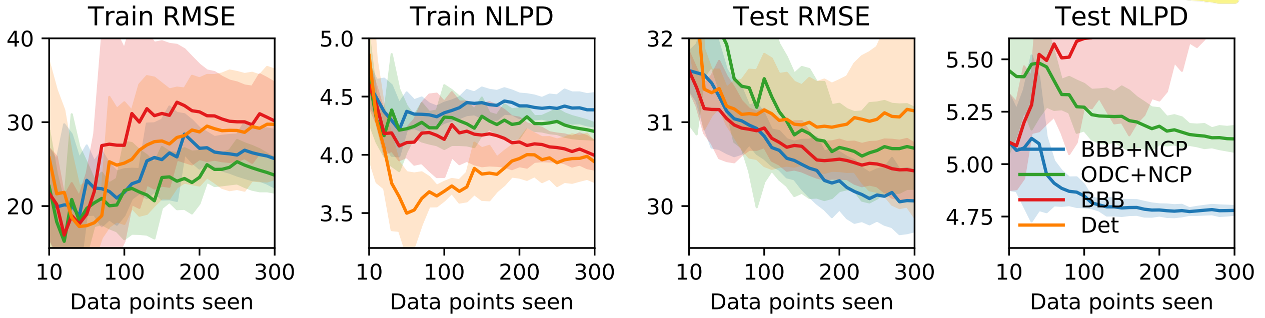


Figure 4: Active learning on the flights data set. The models trained with NCP achieve significantly lower negative log predictive density (NLPD) on the test set, and Bayes by Backprop with NCP achieves the lowest root mean squared error (RMSE). The test NLPD for the baseline models diverges as they overfit to the visible data points. Plots show mean and std over 10 runs.

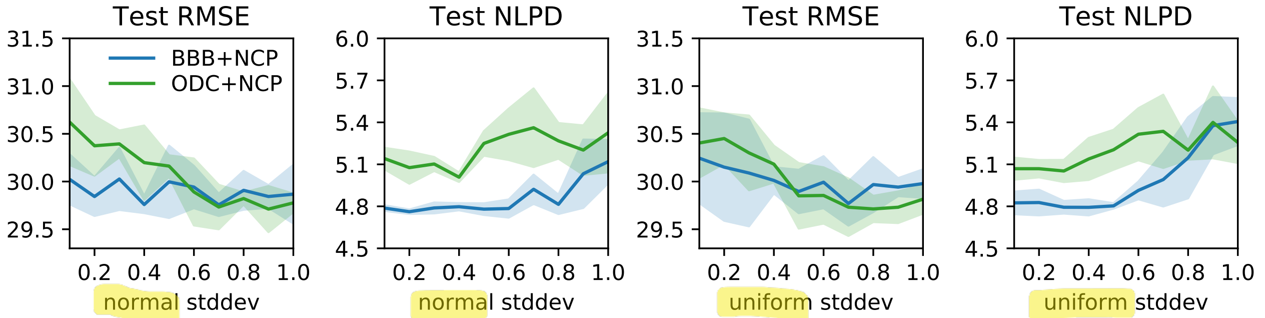


Figure 5: Robustness to different noise patterns. Plots show the final test performance on the flights active learning task (mean and stddev over 5 seeds). Lower is better. NCP is robust to the choice of input noise and improves over the baselines in all settings (compare Figure 4).

We focused on active learning for regression tasks, where uncertainty is crucial for determining which data points to select next. NCPs are only one form of a data prior, designed to encourage uncertainty on OOD inputs. In future work, it would be interesting to apply NCPs to alternative settings where uncertainty is important, such as image classification (using correlated noise noise, such as mixup (Zhang et al., 2018)) and learning with sparse or missing data. Priors in data space can easily capture properties such as periodicity or spatial invariance, and they may provide a scalable alternative to Gaussian process priors.

Acknowledgements We thank Balaji Lakshminarayanan, Jascha Sohl-Dickstein, Matthew D. Hoffman, and Rif Saurous for their comments.

Table 1: Performance on all 700K data points of the flights data set. While uncertainty estimates are not necessary when a large data set that is similar to the test data set is available, it shows that our method scales easily to large data sets.

Model	**NLPD**	**RMSE**
gPoE (Deisenroth & Ng 2015)	8.1	—
SAVIGP (Bonilla et al. 2016)	5.02	—
SVI GP (Hensman et al. 2013)	—	32.60
HGP (Ng & Deisenroth 2014)	—	27.45
MF (Lakshminarayanan et al. 2016)	4.89	26.57
BBB	**4.38**	**24.59**
BBB+NCP	**4.38**	**24.71**
ODC+NCP	**4.38**	**24.68**

REFERENCES

Aboleth Developers. Aboleth. https://github.com/data61/aboleth, 2017.

G. An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1): 108–116, 1995.

D. M. Blei. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232, 2014.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

W. L. Buntine and A. S. Weigend. Bayesian backpropagation. *Complex Systems*, 5(6):603–643, 1991.

R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold gaussian processes for regression. *arXiv preprint arXiv:1402.5876*, 2014.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Neural Information Processing Systems (NIPS)*, 2004.

M. P. Deisenroth and J. W. Ng. Distributed gaussian processes. *arXiv preprint arXiv:1502.02843*, 2015.

J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, and J. Hopfield. Large automatic learning, rule extraction, and generalization. *Complex Systems*, 1(5):877–922, 1987.

S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. Uncertainty decomposition in bayesian neural networks with latent variables. In *International Conference on Machine Learning*, 2018.

J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

D. Flam-Shepherd, J. Requeima, and D. Duvenaud. Mapping gaussian process priors to bayesian neural networks. In *NIPS Workshop*, 2017.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

Y. Gal, R. Islam, and Z. Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

S. Ghosh and F. Doshi-Velez. Model selection in bayesian neural networks via horseshoe priors. *arXiv preprint arXiv:1705.10388*, 2017.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

E. T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2372–2379. IEEE, 2009.

A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, pages 1097–1105, 2012.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18 (1):430–474, 2017.

B. Lakshminarayanan, D. M. Roy, and Y. W. Teh. Mondrian forests for large-scale regression when uncertainty matters. In *Artificial Intelligence and Statistics*, pages 1478–1487, 2016.

J. Lampinen and A. Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274, 2001.

M. Lázaro-Gredilla and A. R. Figueiras-Vidal. Marginalized neural network mixtures for large-scale regression. *IEEE transactions on neural networks*, 21(8): 1345–1351, 2010.

K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

X. Li and Y. Guo. Adaptive active learning for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 859–866. IEEE, 2013.

S. Liang, Y. Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2018.

D. V. Lindley et al. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

C. Louizos, K. Ullrich, and M. Welling. Bayesian compression for deep learning. In *Neural Information Processing Systems*, pages 3290–3300, 2017.

A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 2013.

D. J. MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4): 590–604, 1992a.

D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992b.

A. Malinin and M. Gales. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501*, 2018.

K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

A. K. McCallumzy and K. Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273, 2013.

P. Müller and D. R. Insua. Issues in bayesian analysis of neural network models. *Neural Computation*, 10(3): 749–770, 1998.

R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IIEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1998.

N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

P. Schulam and S. Saria. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems*, pages 748–756, 2015.

B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

S. Sun, G. Zhang, J. Shi, and R. Grosse. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

K.-K. Sung. Learning and example selection for object and pattern detection. *MIT A.I. Memo No. 1521*, 1994.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems*, 2014.

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*, pages 2818–2826, 2016.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

D. Tran, D. Mike, M. van der Wilk, and D. Hafner. Bayesian layers: A module for neural network uncertainty. *arXiv preprint arXiv:1812.03973*, 2018.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2017.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

A OOD CLASSIFIER MODEL WITH NCP

We showed how to apply NCP to a Bayesian neural network model that captures function uncertainty in a belief over parameters. An alternative approach to capture uncertainty is to make explicit predictions about whether an input is OOD. There is no belief over weights in this model. Figure 2b shows such a mixture model via a binary variable o,

$$o \sim \text{Bernoulli}(\pi(x,\theta)) \qquad y \sim \begin{cases} \text{Normal}(\mu(x,\theta), \sigma^2(x,\theta)) & \text{if } o = 0 \\ \text{Normal}(\mu_y, \sigma_y^2) & \text{if } o = 1, \end{cases} \tag{8}$$

where $p(o = 1 \mid x)$ is the OOD probability of x. If $o = 0$ ("in distribution"), the model outputs the neural network prediction. Otherwise, if $o = 1$ ("out of distribution"), the model uses a fixed output prior. The neural network weights θ are estimated using a point estimate, so we do not maintain a belief distribution over them.

The classifier prediction $p(o \mid x, \theta)$ captures uncertainty in this model. We apply the NCP $p(o \mid \tilde{x}, \theta) = \delta(o = 1|\tilde{x}, \theta)$ to this variable, which assumes noised-up inputs to be OOD. During training on the data set, $\{x, y\}$ and $o = 0$ are observed, as training data are in-distribution by definition. Following Equation 2, the loss function is

$$\begin{aligned} \mathcal{L}(\theta) &= D_{\text{KL}}[p_{\text{train}}(y \mid x) \parallel p_{\text{model}}(y \mid x, o = 0, \theta)] + D_{\text{KL}}[p_{\text{prior}}(\tilde{o} \mid \tilde{x}) \parallel p_{\text{model}}(\tilde{o} \mid \tilde{x}, \theta)] \\ &= -\ln p(y, o = 0 \mid x, \theta) - \ln p(y, o = 1 \mid \tilde{x}, \theta) \\ &= -\ln \text{Normal}(y \mid \mu(x,\theta), \sigma^2(x,\theta)) - \ln \text{Bernoulli}(0 \mid \pi(x,\theta)) \underbrace{- \ln \text{Bernoulli}(1 \mid \pi(\tilde{x},\theta))}_{\text{NCP loss}}. \end{aligned} \tag{9}$$

Analogously to the Bayesian neural network model in Section 3, we can either set μ_y, σ_y^2 manually or use the neural network prediction for potentially improved generalization. In our experiments, we implement the OOD classifier model using a single neural network with two output layers that parameterize the Gaussian distribution and the binary distribution.

B DERIVING VARIATIONAL INFERENCE WITH NCP

In Section 3, we described a variational inference objective with NCP which takes the log-likelihood term and adds a forward KL-divergence from the mean prior to the model mean. To derive this:

$$\begin{aligned} \mathrm{E}_{p(x,y)}\big[\ln p(y \mid x)\big] &= \mathrm{E}_{p(x,y)}\Big[\ln \int p(y \mid x, \theta) p(\theta) \frac{q(\theta)}{q(\theta)}\, d\theta\Big] \\ &\geq \mathrm{E}_{p(x,y)}\Big[\int q(\theta) \ln p(y \mid x, \theta) \frac{p(\theta)}{q(\theta)}\, d\theta\Big] \\ &= \mathrm{E}_{p(x,y)}\big[\mathrm{E}_{q(\theta)}[\ln p(y \mid x, \theta)] - D_{\text{KL}}[q(\theta) \parallel p(\theta)]\big] \\ &= \mathrm{E}_{p(x,y)}\big[\mathrm{E}_{q(\theta)}[\ln p(y \mid x, \theta)] - \mathrm{E}_{p(\tilde{x}|x)}[D_{\text{KL}}[q(\theta) \parallel p(\theta)]]\big] \\ &\approx \mathrm{E}_{p(x,y)}\big[\mathrm{E}_{q(\theta)}[\ln p(y \mid x, \theta)] - \mathrm{E}_{p(\tilde{x}|x)}[D_{\text{KL}}[q(\mu(\tilde{x})) \parallel p(\mu(\tilde{x}) \mid x)]]\big], \end{aligned} \tag{10}$$

where $p(\mu(\tilde{x})) = \int \mu(\tilde{x}, \theta) p(\theta)\, d\theta$ and $q(\mu(\tilde{x})) = \int \mu(\tilde{x}, \theta) q(\theta)\, d\theta$ are the distributions of the predicted mean induced by the weight beliefs. As a result, instead of specifying a prior in weight space, we can specify a prior in output space.

Above, we reparameteterized the KL in weight space as a KL in output space; by the change of variables, this is equivalent if the mapping $\mu(\cdot, \theta)$ is continuous and 1-1 with respect to θ. This assumption does not hold for neural nets as multiple parameter vectors can lead to the same predictive distribution, thus the approximation above. A compact reparameterization of the neural network (equivalence class of parameteters) would make this an equality.

Note that the derivation uses the opposite direction of the KL divergence than what we use in the main text. The forward KL divergence we use was originally motivated from maximum likelihood with data augmentation, in which the data prior appears on the left-hand-side of the KL divergence when interpreting maximum likelihood as minimizing the KL divergence from the data distribution to the model. In preliminary experiments, we haven't found that the direction makes a significant difference, but this requires future investigation.

C ROBUSTNESS EXPERIMENT ON TOY DATASET

See Figure 6.

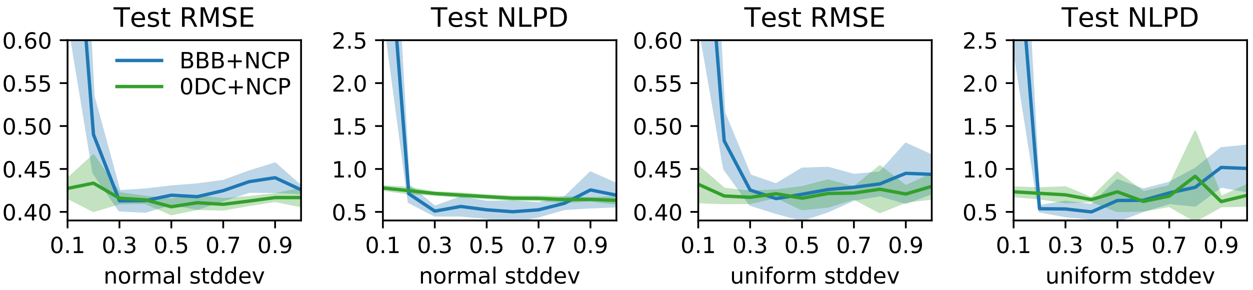


Figure 6: Robustness to different noise patterns. Plots show the final test performance on the low-dimensional active learning task (mean and stddev over 5 seeds). Lower is better. The baseline performances are RMSE: BBB (0.75 ± 0.31), Det (1.46 ± 0.64) and NLPD: BBB (10.29 ± 8.05), Det $(1.3 \times 10^8 \pm 1.7 \times 10^8)$. NCP works with both Gaussian and uniform input noise ϵ and is robust to σ_x^2.

D RELATED ACTIVE LEARNING WORK

Active learning is often employed in domains where data is cheap but labeling is expensive, and is motivated by the idea that not all data points are equally valuable when it comes to learning (Settles, 2009; Dasgupta, 2004). Active learning techniques can be coarsely grouped into three categories. Ensemble methods (Seung et al., 1992; McCallumzy and Nigamy, 1998; Freund et al., 1997) generate queries that have the greatest disagreement between a set of classifiers. Error reduction approaches incorporate the select data based on the predicted reduction in classifier error based on information (MacKay, 1992a), Monte Carlo estimation (Roy and McCallum, 2001), or hard-negative example mining (Sung, 1994; Rowley et al., 1998).

Uncertainty-based techniques select samples for which the classifier is most uncertain. Approaches include maximum entropy (Joshi et al., 2009), distance from the decision boundary (Tong and Koller, 2001), pseudo labelling high confidence examples (Wang et al., 2017), and mixtures of information density and uncertainty measures (Li and Guo, 2013). Within this category, the area most related to our work are Bayesian methods. Kapoor et al. (2007) estimate expected improvement using a Gaussian process. Other approaches use classifier confidence (Lewis and Gale, 1994), predicted expected error (Roy and McCallum, 2001), or model disagreement (Houlsby et al., 2011). Recently, Gal et al. (2017) applied a convolutional neural network with dropout uncertainty to images.