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,

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

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

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

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:

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

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.

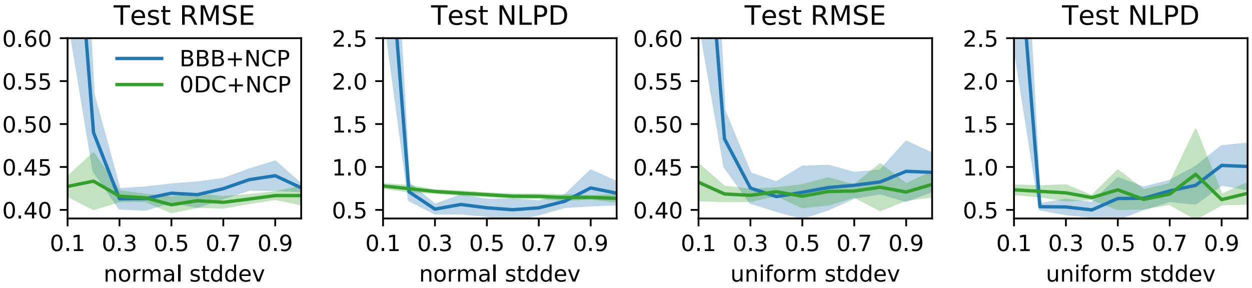

Figure 6: Robustness to different noise patterns. Plots show the final test performance on the low-dimensional active learning task (mean and stddev over 5 seeds). Lower is better. The baseline performances are RMSE: BBB (0.75 ± 0.31), Det (1.46 ± 0.64) and NLPD: BBB (10.29 ± 8.05), Det $(1.3 \times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