Maximizing Overall Diversity for Improved Uncertainty Estimates in Deep Ensembles

Siddhartha Jain,[*1] **Ge Liu,**[*1] **Jonas Mueller,**[2] **David Gifford**[1]
[*]The authors contribute equally, [1]CSAIL,MIT, [2]Amazon Web Services
{sj1, geliu, gifford}@mit.edu, jonasmue@amazon.com

Abstract

The inaccuracy of neural network models on inputs that do not stem from the distribution underlying the training data is problematic and at times unrecognized. Uncertainty estimates of model predictions are often based on the variation in predictions produced by a diverse ensemble of models applied to the same input. Here we describe Maximize Overall Diversity (MOD), an approach to improve ensemble-based uncertainty estimates by encouraging larger overall diversity in ensemble predictions across all possible inputs. We apply MOD to regression tasks including 38 Protein-DNA binding datasets, 9 UCI datasets, and the IMDB-Wiki image dataset. We also explore variants that utilize adversarial training techniques and data density estimation. For out-of-distribution test examples, MOD significantly improves predictive performance and uncertainty calibration without sacrificing performance on test data drawn from same distribution as the training data. We also find that in Bayesian optimization tasks, the performance of UCB acquisition is improved via MOD uncertainty estimates.

Introduction

Model ensembling provides a simple, yet extremely effective technique for improving the predictive performance of arbitrary supervised learners each trained with empirical risk minimization (ERM) (Breiman 1996; Brown 2004). Often, ensembles are utilized not only to improve predictions on test examples stemming from the same underlying distribution as the training data, but also to provide estimates of model uncertainty when learners are presented with out-of-distribution (OOD) examples that may look different than the data encountered during training (Lakshminarayanan, Pritzel, and Blundell 2017; Osband et al. 2016). The widespread success of ensembles crucially relies on the variance-reduction produced by aggregating predictions that are statistically prone to different types of individual errors (Kuncheva and Whitaker 2003). Thus, prediction improvements are best realized by using a large ensemble with many base models, and a large ensemble is also typically employed to produce stable distributional estimates of model uncertainty (Breiman 1996; Papadopoulos, Edwards, and Murray 2001).

Practical applications of massive neural networks (NN) are commonly limited to small ensembles because of the unwieldy nature of these models (Osband et al. 2016; Balan et al. 2015; Beluch et al. 2018). Although supervised learning performance may be enhanced by an ensemble comprised of only a few ERM-trained models, the resulting ensemble-based uncertainty estimates can exhibit excessive sampling variability in low-density regions of the underlying training distribution. Consider the example of an ensemble comprised of five models whose predictions just might agree at points far from the training data by chance. Figure 1 depicts an example of this phenomenon, which we refer to as *uncertainty collapse*, since the resulting ensemble-based uncertainty estimates would indicate these predictions are of high-confidence despite not being supported by any nearby training datapoints.

Unreliable uncertainty estimates are highly undesirable in applications where future input queries may not stem from the same distribution. A shift in input distribution can be caused by sampling bias, covariate shift, and the adaptive experimentation that occurs in bandits, Bayesian optimization (BO), and reinforcement learning (RL) contexts. Here, we propose Maximize Overall Diversity (MOD), a technique to stabilize OOD model uncertainty estimates produced by an ensemble of arbitrary neural networks. The core idea is to consider *all* possible inputs and encourage as much overall diversity in the corresponding model ensemble outputs as can be tolerated without diminishing the ensemble's predictive performance. MOD utilizes an auxiliary loss function and data-augmentation strategy that is easily integrated into any existing training procedure.

Related Work

NN ensembles have been previously demonstrated to produce useful uncertainty estimates for sequential experimentation applications in Bayesian optimization and reinforcement learning (Papadopoulos, Edwards, and Murray 2001; Lakshminarayanan, Pritzel, and Blundell 2017; Riquelme, Tucker, and Snoek 2018). Proposed methods to improve ensembles include adversarial training to enforce smoothness (Lakshminarayanan, Pritzel, and Blundell 2017), and maximizing ensemble output diversity over the training data (Brown 2004). Recent work has proposed regularizers based on augmented out-of-distribution examples, but is primarily specific to classification tasks and non-trivially requires auxiliary generators of OOD examples (Lee et al. 2018) or

existing examples from other classes (Vyas et al. 2018). Another line of related work solely aims at producing better out-of-distribution detectors (Liang, Li, and Srikant 2017; Choi and Jang 2018; Ren et al. 2019).

Our work seeks to improve uncertainty estimates in regression settings, where OOD data can stem from an arbitrary unknown distribution, and robust prediction on OOD data is desired rather than just detection of OOD examples. We propose a simple technique to regularize ensemble behavior over *all* possible inputs that does not require training of additional generator. Consideration of all possible inputs has previously been advocated by (Hooker and Rosset 2012), although not in the context of uncertainty estimation. Pearce et al. (2018) propose a regularizer to ensure an ensemble approximates a valid Bayesian posterior, but their methodology is only applicable to homoskedastic noise unlike ours. Hafner et al. (2018) also aim to control Bayesian NN output-behavior beyond the training distribution, but our methods do not require the Bayesian formulation they impose and can be applied to arbitrary NN ensembles, which are one of the most straightforward methods used for quantifying NN uncertainty (Papadopoulos, Edwards, and Murray 2001; Lakshminarayanan, Pritzel, and Blundell 2017; Riquelme, Tucker, and Snoek 2018). Malinin and Gales (2018) focus on incorporating distributional uncertainty into uncertainty estimates via an additional prior distribution, whereas our focus is on improving model uncertainty in model ensembles.

Methods

We consider standard regression, assuming continuous target values are generated via $Y = f(X) + \epsilon$ with $\epsilon \sim N(0, \sigma_X^2)$, such that σ_X may heteroscedastically depend on feature values X. Given a limited training dataset $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$, where $x_n \sim P_{in}$ specifies the underlying data distribution from which the *in-distribution* examples in the training data are sampled, our goal is to learn an ensemble of M neural networks that accurately models both the underlying function $f(x)$ as well as the uncertainty in ensemble estimates of $f(x)$. Of particular concern are scenarios where test examples x may stem from a different distribution $P_{out} \neq P_{in}$, which we refer to as *out-of-distribution* (OOD) examples. As in (Lakshminarayanan, Pritzel, and Blundell 2017), each network m (with parameters θ_m) in our NN ensemble outputs both an estimated mean $\mu_m(x)$ to predict $f(x)$ and an estimated variance $\sigma_m^2(x)$ to predict σ_x^2, and the per network loss function $L(\theta_m; x_n, y_n) = -\log p_{\theta_m}(y_n|x_n)$, is chosen as the negative log-likelihood (NLL) under the Gaussian assumption $y_n \sim N(\mu_m(x_n), \sigma_m^2(x_n))$. While traditional bagging provides different training data to each ensemble member, we simply train each NN using the entire dataset, since the randomness of separate NN-initializations and SGD-training suffice to produce comparable performance to bagging of NN models (Lakshminarayanan, Pritzel, and Blundell 2017; Lee et al. 2015; Osband et al. 2016).

Following (Lakshminarayanan, Pritzel, and Blundell 2017), we estimate $P_{Y|X=x}$ (and NLL with respect to the ensemble) by treating the aggregate ensemble output as a single Gaussian distribution $N(\bar{\mu}(x), \bar{\sigma}^2(x))$. Here, the ensemble-estimate of $f(x)$ is given by $\bar{\mu}(x) = \text{mean}(\{\mu_m(x)\}_{m=1}^M)$, and the uncertainty in the target value is given by $\bar{\sigma}^2(x) = \sigma_{\text{eps}}^2(x) + \sigma_{\text{mod}}^2(x)$ based on noise-level estimate $\sigma_{\text{eps}}^2(x) = \text{mean}(\{\sigma_m^2(x)\}_{m=1}^M)$ and model uncertainty estimate $\sigma_{\text{mod}}^2(x) = \text{variance}(\{\mu_m(x)\}_{m=1}^M)$. While we focus on Gaussian likelihoods for simplicity, our proposed methodology is applicable to general parametric conditional distributions.

Maximizing Overall Diversity (MOD)

Assuming $X \in \mathcal{X}, Y \in [-C, C]$ have been scaled to bounded regions, MOD encourages higher ensemble diversity by introducing an auxiliary loss that is computed over augmented data sampled from another distribution Q_X. Like P_{in}, Q_X is also defined over the input feature space, but differs from the underlying training data distribution and instead describes OOD examples that could be encountered at test-time. The underlying population objective we target is

$$\min_{\theta_1,\ldots,\theta_M} L_{in} - \gamma L_{out} \quad \text{where}$$
$$L_{in} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{P_{in}}[L(\theta_m, x, y)] \tag{1}$$
$$L_{out} = \mathbb{E}_Q[\sigma_{\text{mod}}^2(x)]$$

with L as the original supervised learning loss function (e.g. NLL), and a user-specified penalty $\gamma > 0$. Since NLL entails a proper-scoring rule (Lakshminarayanan, Pritzel, and Blundell 2017), minimizing the above objective with a sufficiently small value of γ will ensure the ensemble seeks to recover $P_{Y|X=x}$ for inputs x that lie in the support of the training distribution P_{in} and otherwise output large model uncertainty for OOD x that lie outside this support. As it is difficult in most applications to specify how future OOD examples may look, we aim to ensure our ensemble outputs high uncertainty estimates for any possible P_{out} by taking the entire input space into consideration. To account for any possible OOD distribution, we simply pick Q_X as the uniform distribution over $\mathcal{X}$, the bounded region of all possible inputs x. This choice is motivated by Theorem 1 below, which states that the uniform distribution most closely approximates all possible OOD distributions in the minimax sense.

Theorem 1 *The uniform distribution Q_X equals: $\arg\min_{Q\in\mathcal{P}} \max_{P_{out}\in\mathcal{P}} KL(P||Q)$ where for discrete $\mathcal{X}$, $\mathcal{P}$ denotes the set of all distributions, and for continuous $\mathcal{X}$, $\mathcal{P}$ is the set of all distributions with density functions that are bounded within some interval $[a, b]$.*

Proof For the discrete case with $|\mathcal{X}| = N$: let P_{out}, Q have corresponding pmf p, q, so $KL(P_{out}||Q) = \sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}} p(x)\log q(x)$. When Q is the uniform distribution, the worst case P_{out} is one that puts all its mass on a single point x, which corresponds to $KL(P_{out}||Q) = \log N$. For any non-uniform Q': there exists x' where $q'(x') < q(x') = 1/N$. Thus for P'_{out} which puts all its mass on x', we have $KL(P'_{out}||Q') > \log N$. The proof for the continuous case is similar. ■

In practice, we approximate L_{in} using the average loss over the training data as in ERM, and train each θ_m with

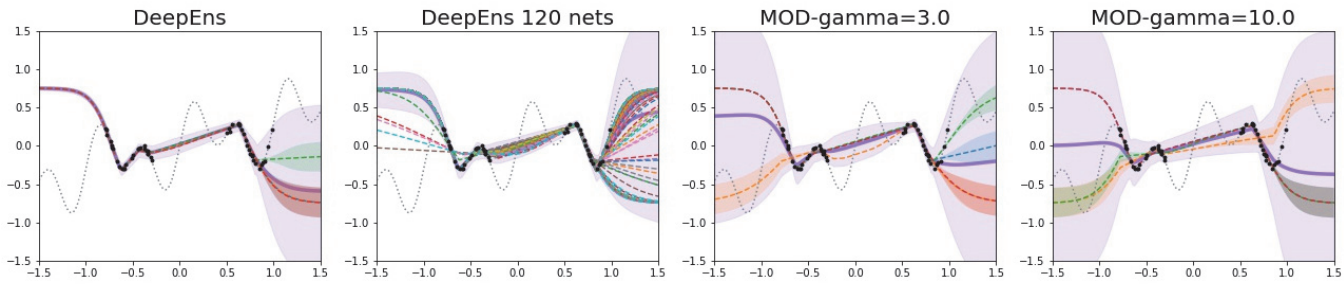

Figure 1: Regression on synthesized data with 95% confidence intervals (CI). The training examples are depicted as black dots and the ground-truth function as a grey dotted line. The predicted conditional mean and CI from individual networks are drawn in colored dashed lines/bands and the overall ensemble conditional mean/CI are depicted via the smooth purple line/band.

respect to its contribution to this term independently of the others as in bagging. To approximate L_{out}, we similarly utilize an empirical average based on augmented examples $\{x_j\}_{j=1}^{K}$ sampled uniformly throughout the feature space $\mathcal{X}$. Uniformly sampling from the input space takes constant time to compute. We expect only a marginal increase in terms of training time since the computation of back-propagation is largely parallelized and thus an increase in minibatch size would only cause an increase in memory consumption rather than computation time. The formal MOD procedure is detailed in Algorithm 1. We advocate selecting γ as the largest value for which estimates of L_{in} (on held-out validation data) do not indicate worse predictive performance. This strategy naturally favors smaller values of γ as the sample size N grows, thus resulting in lower model uncertainty estimates (with $\gamma \to 0$ as $N \to \infty$ when P_{in} is supported everywhere and our NN are universal approximators).

We also experiment with an alternative choice of Q_X being the uniform distribution over the finite training data (i.e. $q(x) = 1/N \ \forall x \in \mathcal{D}$ and = 0 otherwise). We call this alternative method MOD-in, and note its similarity to the diversity-encouraging penalty proposed by (Brown 2004), which is also measured over the training data. Note that MOD in contrast considers Q_X to be uniformly distributed over all possible test inputs rather than only the training examples. Maximizing diversity solely over the training data may fail to control ensemble behavior at OOD points that do not lie near any training example, and thus fail to prevent uncertainty collapse.

Maximizing Reweighted Diversity (MOD-R)

Aiming for high-quality OOD uncertainty estimates, we are mostly concerned with regularizing the ensemble-variance around points located in low density regions of the training data distribution. To obtain a simple estimate that intuitively reflects the inverse of the local density of P_{in} at a particular set of feature values, one can compute the feature-distance to the nearest training data points (Papernot and McDaniel 2018). Under this perspective, we want to encourage greater model uncertainty for the lowest density points that lie furthest from the training data. Commonly used covariance kernels for Gaussian Process regressors (e.g. radial basis functions) explicitly enforce a high amount of uncertainty on points that lie far from the training data. As calculating the distance of each point to the entire training set may be undesirably inefficient for large datasets, we only compute the distance of our augmented data to a current minibatch $\mathcal{B}$ during training. Specifically, we use these distances to compute the following:

$$\tilde{w}_b = \frac{\sum_{i=1}^{k} ||\tilde{x}_b - x_i^b||_2^2}{\max_b \sum_{i=1}^{k} ||\tilde{x}_b - x_i^b||_2^2} \tag{2}$$

where $\tilde{w}_b$ are weights for each of the augmented points $\tilde{x}_b$, and x_i^b $(i = 1, \ldots, k)$ are members of the minibatch $\mathcal{B}$ that are the k nearest neighbors of $\tilde{x}_b$. Throughout this paper, we use $k = 5$.

The $\tilde{w}_b$ are thus inversely related to a crude density estimate of the training distribution P_{in} evaluated at each augmented sample $\tilde{x}_b$. Rather than optimizing the loss L_{out} which uniformly weights each augmented sample (as done in Algorithm 1), we can instead form a weighted loss computed over the minibatch of augmented samples as: $\sum_{b=1}^{|\mathcal{B}|} \tilde{w}_b \cdot \sigma^2_{\text{mod}}(\tilde{x}_b)$ which should increase the model uncertainty for augmented inputs proportionally to their distance from the training data. We call this variant of our methodology with augmented input reweighting MOD-R.

Maximizing Overall Diversity with Adversarial Optimization (MOD-Adv)

We also consider another variant of MOD that utilizes adversarial training techniques. Here, we maximize the variance on relatively over-confident points in out-of-distribution regions, which are likely to comprise worst-case P_{out}. Specifically, we formulate a maximin optimization for the MOD penalty $\max_{\Theta} \min_{x} \sigma^2_{\text{mod}}(x)$, and thus the full training objective becomes $\min_{\theta_1,\ldots,\theta_M} L_{in} - \gamma \cdot \min_x \sigma^2_{\text{mod}}(x)$. We call this variant MOD-Adv. In practice, we obtain the augmented points by taking a single gradient step in the direction of lower variance (σ^2_{mod}), starting from uniformly sampled points. The extra gradient step can double the computation time compared to MOD. The full algorithm is given in Algorithm 1. Note that MOD-Adv is different than the traditional adversarial training in two aspects: first it takes a gradient step with

regard to the model uncertainty measurement (the variance of ensemble mean prediction) instead of with regard to the predicted score of another class; second, the adversarial step is taken starting from a uniformly sampled example instead of a training example. We apply MOD-Adv to only regression tasks with continuous features since it is more natural to apply gradient descent on them.

Algorithm 1 MOD Training Procedure (+ Variants)

Input: Training data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, penalty $\gamma > 0$, batch-size $|\mathcal{B}|$
Output: Parameters of ensemble of M neural networks θ_1,...,θ_M
Initialize θ_1,...,θ_M randomly, initialize $w_b = 1$ for $b = 1, \ldots, |\mathcal{B}|$
repeat
  Sample minibatch from training data:$\{(x_b, y_b)\}_{b=1}^{|\mathcal{B}|}$
  Sample $|\mathcal{B}|$ augmented inputs $\widetilde{x}_1$,..., $\widetilde{x}_B$ uniformly at random from $\mathcal{X}$
  if MOD-Adv **then**
    $\tilde{x}_b \leftarrow \widetilde{x}_b - \alpha_{adv} \cdot \nabla_{\widetilde{x}_b} \sigma^2_{\text{mod}}(\widetilde{x}_b) \;\; \forall 1 \leq b \leq |\mathcal{B}|$
  for m = 1,. . . , M **do**
    if MOD-R **then** $w_b = \tilde{w}_b$ (defined in equation (2))
    Update θ_m via SGD with gradient
$$= \frac{1}{|\mathcal{B}|} \nabla_{\theta_m} \left[\sum_{b=1}^{|\mathcal{B}|} L(\theta_m; (x_b, y_b)) - \gamma \sum_{b=1}^{|\mathcal{B}|} w_b \cdot \sigma^2_{\text{mod}}(\widetilde{x}_b) \right]$$
until iteration limit reached

Experiments

Baseline Methods

Here, we evaluate various alternative strategies for improving model ensembles. All strategies are applied to the same base NN ensemble, which is taken to be the Deep Ensembles (DeepEns) model of (Lakshminarayanan, Pritzel, and Blundell 2017) previously described in **Methods**.

Deep Ensembles with Adversarial Training (DeepEns+AT) (Lakshminarayanan, Pritzel, and Blundell 2017) used this strategy to improve their basic DeepEns model. The idea is to adversarially sample inputs that lie close to the training data but on which the NLL loss is high (assuming they share the same label as their neighboring training example). Then, we include these adversarial points as augmented data when training the ensemble, which smooths the function learned by the ensemble. Starting from training example x, we sample augmented datapoint $x' = x + \delta \text{sign}(\nabla_x L(\theta, x, y))$ with the labels for x' assumed to be the same as that for the corresponding x. L here denotes the NLL loss function, and the values for hyperparameter δ that we search over include 0.05, 0.1, 0.2.

Negative Correlation (NegCorr) This method from (Liu and Yao 1999; Shui et al. 2018) minimizes the empirical correlation between predictions of different ensemble members over the training data. It adds a penalty to the loss of the form $\sum_m [(\mu_m(x) - \bar{\mu}(x)) \cdot \sum_{m' \neq m} (\mu_{m'}(x) - \bar{\mu}(x))]$ where $\mu_m(x)$ is the prediction of the mth ensemble member and $\bar{\mu}(x)$ is the mean ensemble prediction. This penalty is weighted by a user-specified penalty γ, as done in our methodology.

Experiment Details

All experiments were run on Nvidia TitanX 1080 Ti and Nvidia TitanX 2080 Ti GPUs with PyTorch version 1.0. Unless otherwise indicated, all p-values were computed using a single tailed paired t-test per dataset, and the p-values are combined using Fisher's method to produce an overall p-value across all datasets in a task. All hyperparameters – including learning rate, ℓ_2-regularization, γ for MOD/Negative Correlation, and adversarial training δ – were tuned based on validation set NLL. In every regression task, the search for hyperparameter γ was over the values 0.01, 0.1, 1, 5, 10, 20, 50. For MOD-Adv, we search for δ over 0.2,1.0,3.0,5.0 for UCI and 0.1,0.5,1 for the image data.

Univariate Regression

We first consider a one-dimensional regression toy dataset that is similar to the one used by (Blundell et al. 2015). We generated training data from the function:

$$y = 0.3x + 0.3\sin(2\pi x) + 0.3\sin(4\pi x) + \epsilon$$
$$\text{with } \epsilon \sim \mathcal{N}(0, 0.02)$$

Here, the training data only contain samples drawn from two limited-size regions. Using the standard NLL loss as well as the auxiliary MOD penalty, we train a deep ensemble with 4 neural networks of identical architectures consisting of 1-hidden layer with 50 units, ReLU activation, two sigmoid outputs to estimate the mean and variance of $P_{Y|X=x}$, and L2 regularization. To depict the improvement gained by simply adding ensemble members, we also train an ensemble of 120 networks with same architecture. Figure 1 shows the predictions and 95% confidence interval of the ensembles. MOD is able to produce more reliable uncertainty estimates on the lefthand regions that lack training data, whereas standard deep ensembles exhibit uncertainty collapse, even with many networks. MOD also properly inflated the predictive uncertainty in the center region where no training data is found. Using a smaller $\gamma = 3$ in MOD ensures the ensemble predictive performance remains strong for in-distribution inputs that lie near the training data and the ensemble exhibits adequate levels of certainty around these points. While the larger $\gamma = 10$ value leads to overly conservative uncertainty estimates that are large everywhere, we note the mean of the ensemble predictions remains highly accurate for in-distribution inputs.

Protein Binding Microarray Data

We next study scientific data with discrete features by predicting Protein-DNA binding. This is a collection of 38 different microarray datasets, each of which contains measurements of the binding affinity of a single transcription factor (TF) protein against all possible 8-base DNA sequences (Barrera et al. 2016). We consider each dataset as a separate task with

Y taken to be the binding affinity scaled to the interval [0,1] and X the one-hot embedded DNA sequence. As we ignore reverse-complements, there are $\sim 32,000$ possible values of X.

Regression We trained a small ensemble of 4 neural networks with the same architecture as in the previous experiments. We consider 2 different OOD test sets, one comprised of the sequences with top 10% Y-values and the other comprised of the sequences with more than 80% of the position in X being G or C (GC-content). For each OOD set, we use the remainder of the sequences as corresponding in-distribution set. We separate them into extremely small training set (300 examples) and validation set (300 examples), and use the rest as in-distribution test set. We compare MOD along with 3 alternative sampling distribution (MOD-in, MOD-R, and MOD-Adv) against the 3 baselines previously mentioned. We search over 0,1e-3,0.01,0.05,0.1 for $l2$ penalty and 0.01 for learning rate.

Table 1 and Appendix Table 1 shows mean OOD and in-distribution performance across 38 TFs (averaged over 10 runs using random data splits and NN initializations). MOD methods have significantly improved performance on all metrics and OOD setups compared to DeepEns/DeepEns+AT, both in terms of # of TF outperforming and overall p-value and is on par with DeepEns+AT on in-Distribution. The reweighting scheme (MOD-R) further improved the performance on top 10% Y-value OOD set up. Figure 2 shows the calibration curve on two of the TFs where the deep ensembles are over-confident on top 10% Y-value OOD examples. MOD-R and MOD improve the calibration results by significant margin compared to most of the baselines.

Bayesian Optimization Next, we compared how the MOD, MOD-R, and MOD-in ensembles performed against the DeepEns, DeepEns+AT, and NegCorr ensembles in 38 Bayesian optimization tasks using the same protein binding data (Hashimoto, Yadlowsky, and Duchi 2018). For each TF, we performed 30 rounds of DNA-sequence acquisition, acquiring batches of 10 sequences per round in an attempt to maximize binding affinity. We used the *upper confidence bound* (UCB) as our acquisition function (Chen et al. 2017), ordering the candidate points via $\bar{\mu}(x) + \beta \cdot \sigma_{\text{mod}}(x)$ (with UCB coefficient $\beta = 1$).

At every acquisition iteration, we randomly held out 10% of the training set as the validation set and chose the γ penalty (for MOD, MOD-in, MOD-R, and NegCorr) that produced the best validation NLL (out of choices: 0, 5, 10, 20, 40, 80). The stopping epoch is chosen based on the validation NLL not increasing for 10 epochs with an upper limit of 30 epochs. Optimization was done with a learning rate of 0.01, L2 penalty of 0.01 and used the Adam optimizer. For each of the 38 TFs, we performed 20 Bayesian optimization runs with different seed sequences (same seeds used for all the methods) and using 200 points randomly sampled from the bottom 90% of Y values as are initial training set.

We evaluated on the metric of simple regret $r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [1,T]} f(x_t)$ (second term in the subtraction quantifies the best point acquired so far and the first term is the global best). The results are presented in Table 2. MOD outperforms all other methods in both number of TFs with better regret and the combined p-value. MOD-R is also strong outperforming all other methods except MOD with respect to which is about equivalent in terms of statistical significance. Figure 3 shows r_T for the TFs *OVOL2* and *HESX1*, a task in which MOD and MOD-R outperform the other methods.

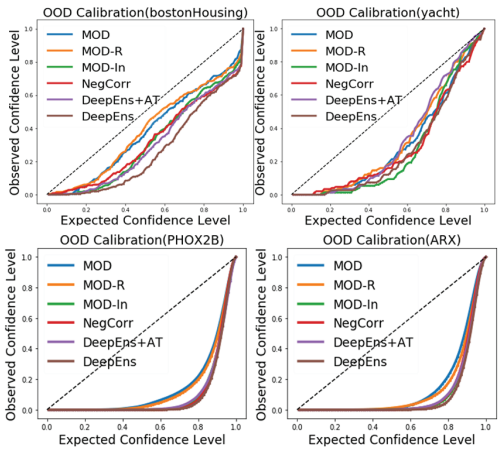

Figure 2: Calibration curves for regression models trained on two of the UCI datasets (top) and two DNA TF binding datasets (bottom). A perfect calibration curve should lie on the diagonal, and an over-confident model has calibration curve where the model expected confidence level is higher than observed confidence level (below diagonal). MOD-R and MOD significantly improve the over-confident predictions from the deep ensembles trained without augmentation loss.

UCI Regression Datasets

We next experimented with 9 real world datasets with continuous inputs in some applicable bounded domain. We follow the experimental setup that (Lakshminarayanan, Pritzel, and Blundell 2017) and (Hernández-Lobato et al. 2017) used to evaluate deep ensembles and deep Bayesian regressors. We split off all datapoints whose y-values fall in the top 5% as an OOD test set (so datapoints with such large y-values are never encountered during training). We simulate the situation where training set is limited and thus used 40% of the data for training and 10% for validation. The remaining data is used as an in-distribution test set. The analysis is repeated for 10 random splits of the data to ensure robustness. We again use an ensemble of 4 fully-connected neural networks with the same architecture as above and the NLL training loss searching over hyperparameter values: L2 penalty $\in \{0, 0.001, 0.01, 0.05, 0.1\}$, learning rate $\in \{0.0005, 0.001, 0.01\}$. We report the negative log-likelihood (NLL) on both in- and out-of-distribution test sets for ensembles trained via different strategies (including MOD-Adv) and examine the calibration curves.

As shown in Table 3, MOD outperforms DeepEns in 6 out

Table 1: NLL on OOD/in-distribution test set averaged across 38 TFs over 10 replicate runs(See Appendix Table 1 for RMSE). MOD out-performance p-value is the combined p-value of MOD NLL being less than the NLL of the method in the corresponding row. **Bold** indicates best in category and ***bold+italicized*** indicates second best. In case of a tie in the means, the method with lower standard deviation is highlighted.

Methods	Out-of-distribution NLL	(MOD out-performance) # of TFs	p-value	In-distribution NLL	(MOD out-performance) # of TFs	p-value
(OOD as sequences with top 10% binding affinity)						
DeepEns	0.7485±0.124	26	1.7e-05	-0.4266±0.031	32	7.7e-09
DeepEns+AT	0.7438±0.122	25	0.001	-0.4312±0.033	26	0.005
NegCorr	0.7358±0.118	27	0.061	-0.4314±0.032	17	0.761
MOD	**0.7153±0.117**	–	–	-0.4312±0.031	–	–
MOD-R	***0.7225±0.116***	22	0.359	**-0.4325±0.032**	16	0.777
MOD-in	0.7326±0.121	26	0.012	***-0.4317±0.032***	19	0.535
(OOD as sequences with >80% GC content)						
DeepEns	-0.6938±0.052	20	0.022	-0.5649±0.029	34	3.1e-11
DeepEns+AT	**-0.7010±0.041**	23	0.007	**-0.5740±0.027**	21	0.292
NegCorr	-0.6805±0.065	25	0.011	-0.5700±0.026	25	0.017
MOD	***-0.7007±0.047***	–	–	***-0.5729±0.027***	–	–
MOD-R	-0.6959±0.040	24	0.004	-0.5720±0.027	22	0.357
MOD-in	-0.6948±0.054	21	0.103	-0.5711±0.028	22	0.163

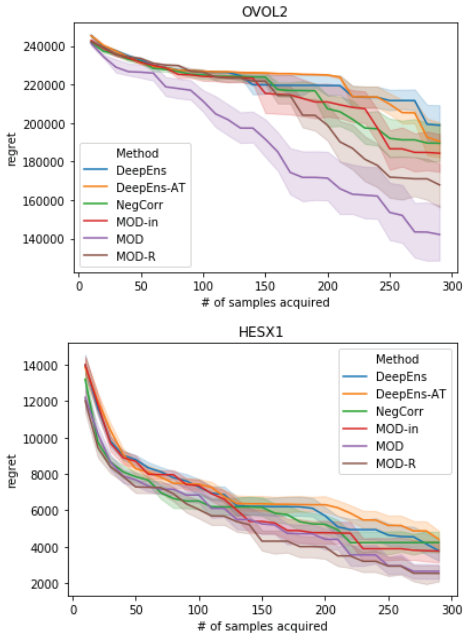

Figure 3: Regret for two Bayesian optimization tasks (averaged over 20 replicate runs). The bands depict 50% confidence intervals, and the x-axis indicates the number of DNA sequences whose binding affinity has been profiled so far.

of the 9 datasets on OOD NLL, and has significant overall p-value compared to all baselines. MOD-Adv ranks top 1 in OOD NLL in terms of averaged ranks across all datasets, showing better robustness than MOD. The MOD loss lead to higher-quality uncertainties on OOD data while also improving in-distribution performance of DeepEns.

Table 2: Regret (r_T) comparison. Each cell shows the number of TFs (out of 38) for which the method in corresponding row outperforms the method in the corresponding column (lower r_T). The number in parentheses is the combined (across 38 TFs) p-value of MOD/-in/-R regret being less than the regret of the method in the corresponding column.

vs	DeepEns	DeepEns+AT	NegCorr
MOD-in	21 (0.111)	21 (0.041)	19 (0.356)
MOD	26 (0.003)	24 (0.004)	20 (0.001)
MOD-R	22 (0.019)	23 (0.007)	22 (0.017)
vs	MOD-in	MOD	MOD-R
MOD-in	–	17 (0.791)	16 (0.51)
MOD	19 (0.002)	–	22 (0.173)
MOD-R	20 (0.052)	14 (0.674)	–

Figure 2 shows the calibration curve on two of the datasets where the basic deep ensembles exhibit over-confidence on OOD data. Note that retaining accurate calibration on OOD data is extremely difficult for most machine learning methods. MOD and MOD-R improve calibration by a significant margin compared to most of the baselines, validating the effectiveness of our MOD procedure.

The selection of γ is critical for MOD, thus we also examine the effect of the choice of γ on the in-distribution performance for the 9 UCI and 38 TF binding regression tasks. As shown in Figure 4, γ generally does not affect or hurt in-distribution NLL until it gets too large at which point it fairly consistently starts hurting it. When γ is selected properly it may even improve the in-distribution slightly as shown in the previous tables.

Table 3: Averaged NLL on out-of-distribution/in-distribution test example over 10 replicate runs for UCI datasets, top 5% samples were heldout as OOD test set (See Appendix Table 2 for RMSE). MOD outperformance p-value is the combined (via Fisher's method) p-value of MOD NLL being less than the NLL of the method in the corresponding column (with p-value per dataset being computed using a paired single tailed t-test). **Bold** indicates best in category and ***bold+italicized*** indicates second best. In case of a tie in the means, the method with the lower standard deviation is highlighted.

Datasets	DeepEns	DeepEns+AT	NegCorr	MOD	MOD-R	MOD-in	MOD-Adv
Out-of-distribution NLL							
concrete	-0.831±0.237	-0.915±0.204	-0.913±0.277	-0.904±0.118	-0.910±0.193	***-0.924±0.188***	**-0.950±0.200**
yacht	-1.597±0.840	-1.762±0.647	**-1.972±0.570**	-1.797±0.437	-1.761±0.578	-1.638±0.663	***-1.948±0.343***
naval-propulsion-plant	-2.580±0.103	-1.380±0.087	-2.618±0.056	**-2.729±0.071**	-2.130±0.069	-2.057±0.055	***-2.629±0.068***
wine-quality-red	0.133±0.132	0.115±0.086	0.113±0.104	0.153±0.107	**0.084±0.072**	***0.085±0.065***	0.217±0.114
power-plant	**-1.734±0.054**	-1.731±0.088	-1.659±0.075	-1.638±0.151	-1.644±0.120	***-1.731±0.050***	-1.669±0.066
protein-tertiary-structure	***1.162±0.231***	1.178±0.158	1.231±0.130	1.197±0.137	1.194±0.214	**1.154±0.132**	1.299±0.252
kin8nm	-1.980±0.053	-1.970±0.093	**-2.036±0.046**	-1.999±0.049	-2.003±0.095	-1.993±0.078	***-2.027±0.085***
bostonHousing	1.591±0.680	1.243±0.690	1.821±0.913	***0.568±0.959***	**0.460±0.648**	0.923±0.733	1.517±0.711
energy	-1.590±0.253	**-1.784±0.153**	-1.718±0.193	-1.736±0.117	-1.741±0.264	-1.733±0.199	***-1.772±0.242***
MOD outperformance p-value	0.002	4.9e-07	0.034	-	1.6e-04	4.6e-05	0.027
In-distribution NLL							
concrete	-1.075±0.094	-1.129±0.084	-1.089±0.102	**-1.155±0.086**	***-1.137±0.132***	-1.090±0.092	-1.047±0.177
yacht	-3.286±0.692	-3.245±0.822	**-3.570±0.166**	-3.500±0.190	-3.461±0.252	-3.339±0.815	***-3.556±0.203***
naval-propulsion-plant	-2.735±0.077	-1.513±0.042	-2.810±0.042	**-2.857±0.067**	-2.297±0.061	-2.238±0.047	***-2.817±0.046***
wine-quality-red	-0.070±0.853	-0.341±0.068	-0.266±0.291	-0.337±0.069	***-0.348±0.045***	**-0.351±0.055**	-0.170±0.505
power-plant	-1.521±0.015	**-1.525±0.018**	-1.524±0.023	-1.523±0.012	-1.522±0.017	***-1.524±0.013***	-1.523±0.016
protein-tertiary-structure	-0.514±0.013	-0.519±0.007	**-0.544±0.012**	-0.533±0.009	-0.532±0.012	-0.529±0.008	***-0.540±0.012***
kin8nm	-1.305±0.016	-1.315±0.020	**-1.334±0.015**	-1.317±0.019	-1.315±0.017	***-1.322±0.015***	-1.314±0.020
bostonHousing	-0.901±0.154	***-0.937±0.144***	-0.656±0.671	**-0.953±0.147**	-0.883±0.188	-0.925±0.180	-0.728±0.376
energy	-2.426±0.151	-2.517±0.098	***-2.620±0.130***	-2.507±0.153	-2.525±0.098	-2.522±0.129	**-2.638±0.137**
MOD outperformance p-value	2.4e-08	6.9e-11	0.116	-	8.9e-06	2.2e-06	0.046

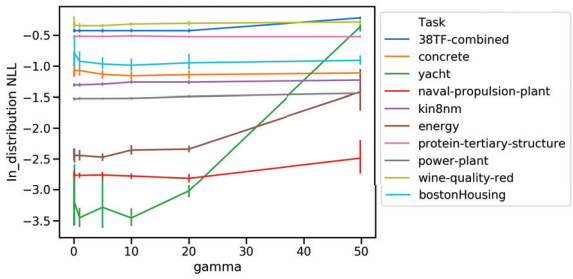

Figure 4: The effect of different γ on in-distribution test performance (NLL).

Age Prediction from Images

To demonstrate the effectiveness of MOD, MOD-Adv, and MOD-R on high dimensional data, we consider supervised learning with image data. Here, we use a dataset of human images collected from IMDB and Wikimedia and annotated with age and gender information (Rothe, Timofte, and Van Gool 2015). The IMDB/Wiki parts of the dataset consist of 460K+/62K+ images respectively. 28,601 images in the Wiki dataset are males and the rest are females.

In the context of Wiki images, we tried to predict the ages given the image of a person using 2000 images of *males as the training set*. For the OOD dataset, we hold out the oldest 10% of the people as the OOD set. We used the Wide Residual Network architecture (Zagoruyko and Komodakis 2016) with a depth of 4 and a width factor of 2. As before, we used an ensemble of size 4. The search for the optimal γ value was over $0, 2, 5, 10, 20, 40, 80$. The stopping epoch is chosen based on the validation NLL not increasing for 10 epochs with an upper limit of 30 epochs. Optimization was done with a learning rate of 0.001, l2 penalty of 0.001 and used the Adam optimizer. The NLL results are in Table 4 whereas the RMSE results are in the Appendix. Both Maximize Overall Diversity and MOD-Adv get the best results on OOD NLL with the improvement being statistically significant over the other methods. MOD gets an NLL of 1.129 on OOD data, MOD-Adv gets an NLL of 1.155 on OOD, and MOD-R gets 1.185 on OOD. This is in contrast to DeepEns which gets only 1.304 on OOD. Thus both MOD and MOD-R show significant improvements on NLL on the OOD data. In addition, while DeepEns+AT has a better mean *in-distribution* NLL compared to MOD, the focus of this paper is out of distribution uncertainty on which Maximize Overall Diversity and MOD-Adv perform very well. Notably every MOD variant *improves* performance for *both* in and out of distribution. Thus augmenting the loss function with the MOD penalty should not make your model worse.

Conclusion

We have introduced a loss function and data augmentation strategy that helps stabilize distribution uncertainty estimates obtained from model ensembling. Our method increases model uncertainty over the entire input space while simultaneously maintaining predictive performance, which helps mitigate uncertainty collapse that may arise in small model ensembles. We further proposed two variants of our method. MOD-R assesses the distance of an augmented sample from the training distribution and aims to ensure higher model uncertainty in regions with low-density, and MOD-Adv uses

Table 4: Image regression results showing mean performance across 20 randomly seeded runs (along with ± one standard deviation). In-Dist refers to the in-distribution test set. OOD refers to the out of distribution test set. **Bold** indicates best in category and ***bold+italicized*** indicates second best. In case of a tie in means, the lower standard deviation method is highlighted.

Methods	OOD NLL	In-Dist NLL
DeepEns	1.3100 ± 0.2486	-0.2193 ± 0.0207
DeepEns+AT	1.2348±0.1291	**-0.2419±0.0213**
NegCorr	1.1731±0.1978	-0.2286±0.0179
MOD-in	1.2625±0.1961	-0.2301±0.0128
MOD	**1.1294±0.1707**	***-0.2306±0.0148***
MOD-R	1.1847±0.2442	-0.2285±0.0191
MOD-Adv	***1.1547±0.1865***	-0.2305±0.0149

adversarial optimization to improve model uncertainty on relatively over-confident regions more efficiently. Our methods produce improvements to both the in and out of distribution NLL, out of distribution RMSE, and calibration on a variety of datasets drawn from biology, vision, and common UCI datasets. We also showed MOD is useful in hard Bayesian optimization tasks. Future work could develop techniques to generate OOD augmented samples for structured data, as well as applying ensembles with improved uncertainty-awareness to currently challenging tasks such as exploration in reinforcement learning.

References

Balan, A. K.; Rathod, V.; Murphy, K. P.; and Welling, M. 2015. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*.

Barrera, L. A.; Vedenko, A.; Kurland, J. V.; Rogers, J. M.; Gisselbrecht, S. S.; Rossin, E. J.; Woodard, J.; Mariani, L.; Kock, K. H.; Inukai, S.; et al. 2016. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science* 351(6280):1450–1454.

Beluch, W. H.; Genewein, T.; Nürnberger, A.; and Köhler, J. M. 2018. The power of ensembles for active learning in image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

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

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24:123–140.

Brown, G. 2004. *Diversity in neural network ensembles*. Ph.D. Dissertation, University of Birmingham.

Chen, R. Y.; Sidor, S.; Abbeel, P.; and Schulman, J. 2017. UCB exploration via Q-ensembles. *arXiv:1706.01502*.

Choi, H., and Jang, E. 2018. Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*.

Hafner, D.; Tran, D.; Lillicrap, T.; Irpan, A.; and Davidson, J. 2018. Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *arXiv:1807.09289*.

Hashimoto, T. B.; Yadlowsky, S.; and Duchi, J. C. 2018. Derivative free optimization via repeated classification. In *International Conference on Artificial Intelligence and Statistics*.

Hernández-Lobato, J. M.; Requeima, J.; Pyzer-Knapp, E. O.; and Aspuru-Guzik, A. 2017. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. *arXiv:1706.01825*.

Hooker, G., and Rosset, S. 2012. Prediction-focused regularization using data-augmented regression. *Statistics and Computing* 1:237–349.

Kuncheva, L. I., and Whitaker, C. J. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* 51:181–207.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*.

Lee, S.; Purushwalkam, S.; Cogswell, M.; Crandall, D.; and Batra, D. 2015. Why M heads are better than one: Training a diverse ensemble of deep networks. *arXiv:1511.06314*.

Lee, K.; Lee, H.; Lee, K.; and Shin, J. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Liu, Y., and Yao, X. 1999. Ensemble learning via negative correlation. *Neural networks* 12(10):1399–1404.

Malinin, A., and Gales, M. 2018. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, 7047–7058.

Osband, I.; Blundell, C.; Pritzel, A.; and Van Roy, B. 2016. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*.

Papadopoulos, G.; Edwards, P. J.; and Murray, A. F. 2001. Confidence estimation methods for neural networks: A practical comparison. *IEEE Transactions on Neural Networks* 12:1278–1287.

Papernot, N., and McDaniel, P. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*.

Pearce, T.; Zaki, M.; Brintrup, A.; and Neel, A. 2018. Uncertainty in neural networks: Bayesian ensembling. *arXiv preprint arXiv:1810.05546*.

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; DePristo, M. A.; Dillon, J. V.; and Lakshminarayanan, B. 2019. Likelihood ratios for out-of-distribution detection. *arXiv preprint arXiv:1906.02845*.

Riquelme, C.; Tucker, G.; and Snoek, J. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*.

Rothe, R.; Timofte, R.; and Van Gool, L. 2015. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 10–15.

Shui, C.; Mozafari, A. S.; Marek, J.; Hedhli, I.; and Gagne, C. 2018. Diversity regularization in deep ensembles. *arXiv:1802.07881*.

Vyas, A.; Jammalamadaka, N.; Zhu, X.; Das, D.; Kaul, B.; and Willke, T. L. 2018. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 550–564.

Zagoruyko, S., and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.