Maximizing Overall Diversity to Control Out-of-Distribution Behavior of Deep Ensembles

Siddhartha Jain* CSAIL, MIT sj1@mit.edu

Ge Liu* CSAIL, MIT geliu@mit.edu

Jonas Mueller CSAIL, MIT jonasmueller@csail.mit.edu

David K. Gifford CSAIL, MIT gifford@mit.edu

Abstract

Deep neural networks give state of the art performance for a wide array of tasks. Extrapolation of predictions to out of distribution data however is a challenge. Uncertainty estimation for predictions can greatly mitigate the harms of wild extrapolation and can be useful in tasks like Bayesian optimization or reinforcement learning. Due to its simplicity, model ensembling is a popular way of estimating uncertainty on in and out of distribution data. Here, we present a new training objective to better control the out of distribution uncertainty of neural network ensembles. The idea is to encourage larger overall ensemble diversity for all possible inputs that might be encountered. We apply our objective to 38 Protein-DNA binding regression datasets and obtain significant improvements on out of distribution negative log-likelihood and RMSE, without sacrificing any in-distribution performance. Our improved uncertainty estimates also lead to better Bayesian optimization performance on a number of affinity-optimization tasks.

1 Introduction

Model ensembling provides a simple, yet extremely effective technique for improving the predictive performance of arbitrary supervised learners each trained via empirical risk minimization (ERM) [1, 2]. 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 [3, 4]. 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 [5]. 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 [1, 6].

Despite this, practical applications of massive neural networks (NN) are commonly limited to a small ensemble due to the unwieldy nature of these models [4, 7, 8]. Although supervised learning performance may still 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. Such unreliable uncertainty estimates are highly undesirable in applications where future data may not always stem from the same distribution (e.g. due to sampling bias, covariate shift, or the adaptive experimentation that occurs in bandits, Bayesian

*Equal contribution

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

optimization (BO), and reinforcement learning (RL) contexts). Here, we propose a straightforward technique - Maximize Overall Diversity (MOD) - to stabilize the 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 than is easily integrated into any existing training procedure.

Related Work. NN ensembles have been previously demonstrated to produce useful uncertainty estimates, including for BO/RL sequential experimentation applications [6, 3, 9, 4, 10]. Proposed methods to improve ensembles of limited size include adversarial training to enforce smoothness [3] and maximizing ensemble output diversity over the training data [2]. In contrast, our focus is on controlling ensemble behavior over *all* possible inputs, not merely those presented during training. Consideration of all possible inputs has been advocated in the method of data-augmented regression [11], although not in the context of uncertainty estimation. Like our approach, Hafner et al. [12] also aim to control 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 neural network uncertainty [6, 4]. While we primarily consider regression settings here, our ideas can be easily adapted to classification by replacing variance terms with entropy terms; a similar variant that relies on an auxiliary generator network to augment the training data has been recently proposed in [13].

2 Methods

We consider a standard regression setup, 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 the feature values X. Given a limited training dataset $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$ where $X \sim P_{in}$ specifies the underlying data distribution (from which samples are called *in-distribution*), 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 may stem from a different distribution P_{out} (i.e. they may be *out-of-distribution*). As in [3], each network m (with parameters θ_m) in our NN ensemble outputs both an estimated mean $\mu_m(x)$ and variance $\sigma_m^2(x)$, and the per network loss function $L(\theta_m; x_n, y_n) = -\log p_{\theta_m}(y_n|x_n)$, the negative log-likelihood (NLL) under our Gaussianity assumption. 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 [3, 14, 4].

Following [3], 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 $f(X)$ (used in RMSE calculations) is given by $\bar{\mu}(x) = \text{mean}\big(\{\mu_m(x)\}_{m=1}^M\big)$, and the uncertainty in the target value is given by: $\bar{\sigma}^2(x) = \sigma^2_{\text{eps}}(x) + \sigma^2_{\text{mod}}(x)$ based on noise-level estimate: $\sigma^2_{\text{eps}}(x) = \text{mean}\big(\{\sigma_m^2(x)\}_{m=1}^M\big)$ and model uncertainty estimate: $\sigma^2_{\text{mod}}(x) = \text{variance}\big(\{\mu_m(x)\}_{m=1}^M\big)$.

Assuming $X \in \mathcal{X}, Y \in [-B, B]$ have been scaled to bounded regions, MOD encourages higher ensemble diversity by introducing an auxiliary loss that is computed over augmented data uniformly sampled throughout the entire feature-space. The underlying population objective we target is:

$$\min_{\theta_1,\ldots,\theta_M} L_{\text{in}} - \gamma L_{out} \quad \text{where: } L_{\text{in}} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{P_{\text{in}}}[L(\theta_m, x, y)], \;\; L_{out} = \frac{1}{Z}\int_{\mathcal{X}} \sigma^2_{\text{mod}}(x)\, \mathrm{d}x$$

with normalization factor $Z = \int_{\mathcal{X}} \mathrm{d}x$, and user-specified penalty $\gamma > 0$. Since NLL entails a proper-scoring rule [3], 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 to output large model uncertainty for OOD x that lie beyond this support. As it is difficult in most applications to specify how future OOD examples may look, we aim to ensure the ensemble outputs high uncertainty estimates for any possible P_{out} by integrating over the entire input space. In practice, we approximate L_{in} using the average loss over the training data (as in ERM), and train each θ_m with 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 feature space $\mathcal{X}$. 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).

Algorithm 1 MOD training procedure

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

3 Results

Protein Binding Microarray Data. This is a collection of 38 different datasets, each of which contains measurements of the binding affinity of a single transcription factor (TF) protein against all possible 8-base DNA sequences [15]. We consider each dataset as a separate task with Y taken to be the binding affinity (rescaled to [0,1] interval) and X the one-hot embedded DNA sequence (as we ignore reverse-complements, there are $\sim 32,000$ possible values of X).

Regression Performance. We trained a small ensemble of 4 neural networks with identical architecture (1-hidden layer with 50 units, ReLU activation, two sigmoid outputs to estimate the mean and variance of Y, and l2 regularization). We experiment with the case where the training set is extremely small (300 examples) and use a small validation set (300 examples) for tuning γ and other NN hyperparameters (based on validation NLL). The remainder of the possible DNA-sequences form the test-set, and we also considered an alternative OOD test set comprised only of the data with Y-values in the top 10%. We compare MOD against two alternatives based on the same NN ensemble without our augmented-loss L_{out}: training the ensemble in the usual fashion (*Normal*), and training the ensemble with an additional loss $= -\gamma\frac{1}{N}\sum_{i=1}^N \sigma^2_{\text{mod}}(x)$ that attempts to maximize diversity over the training data (MTD), similar to the method proposed in [2].

Table 1 shows OOD and in-distribution performance across 38 TFs (averaged over thirteen runs using random data splits and NN initializations). MOD has significantly improved performance on the OOD test set, while performing roughly similar to the other methods on the in-distribution test examples. Out of the 38 datasets, MOD outperforms the Normal ensemble on 28 in OOD-NLL and on 27 on OOD-RMSE, and outperforms the MTD ensemble on 24 in OOD-NLL and on 22 on OOD-RMSE. Over the in-distribution test data, MOD outperforms the Normal ensemble on 31 (and 36) datasets in NLL (and RMSE), and performs comparably to the MTD ensemble (even though MOD is not designed for in-distribution regularization). Tables 2 and 3 contain the complete results for all 38 datasets.

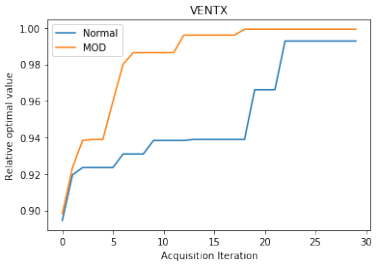


Figure 1: Relative optimal value for one BO task (averaged over 10 replicate BO runs).

	Out-of-distribution		In-distribution	
Method	NLL	RMSE	NLL	RMSE
MOD	**-0.201**	**0.277**	−1.352	**0.157**
Normal	−0.146	0.282	−1.347	0.158
MTD	−0.181	0.278	**-1.352**	0.157

Table 1: Averaged NLL and RMSE on out-of-distribution/in-distribution test example over 13 replicate runs.

Bayesian Optimization. Next, we compared how the MOD ensemble performed against the Normal ensemble in 38 Bayesian optimization tasks using the same protein binding data (see [16]). 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 [10], 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 MOD γ penalty that produced the best validation NLL (out of choices: 5, 10, 20). For each of the 38 TFs, we performed 10 BO runs with different seed sequences (same seeds used between Normal and MOD) of 200 points randomly sampled from the bottom 90% of Y values.

We evaluated on two metrics: relative optimal value $r_T = \frac{\max_{t \in [1,T]} f(x_t)}{\max_{x \in \mathcal{X}} f(x)}$ (numerator quantifies the best point acquired so far and denominator is the global best), and fraction of sequences retrieved out of those with affinity-values in the top 1% [17]. Figure 1 shows r_T for the TF *VENTX*, a task in which MOD clearly outperforms the Normal ensemble (results for other TFs in Figure 2). On most tasks, MOD-BO typically outperforms Normal-BO in the later acquisition rounds and is also able to achieve higher average r_T over all 38 tasks. At the end of 30 acquisition rounds, MOD-BO outperformed Normal-BO for 25 of the 38 tasks (with equal performance on 2 additional tasks). For the fraction of top 1% of points retrieved (on average across 10 BO runs), MOD-BO outperformed Normal-BO for 26 of the 38 TFs (full results in Table 4).

Table 2: NLL and RMSE on out-of-distribution test set for all 38 TF datasets.

	Out-of-distribution NLL			Out-of-distribution RMSE		
TF	MOD	MTD	Normal	MOD	MTD	Normal
PHOX2B	0.03 ± 0.24	0.20 ± 0.15	0.21 ± 0.15	0.34 ± 0.02	0.35 ± 0.02	0.35 ± 0.03
POU6F2	-0.31 ± 0.14	-0.25 ± 0.26	-0.23 ± 0.20	0.28 ± 0.02	0.28 ± 0.02	0.29 ± 0.01
HOXC4	-0.31 ± 0.24	-0.19 ± 0.20	-0.22 ± 0.37	0.27 ± 0.02	0.27 ± 0.02	0.27 ± 0.02
ISX	-0.12 ± 0.17	-0.17 ± 0.14	-0.13 ± 0.15	0.28 ± 0.01	0.27 ± 0.01	0.28 ± 0.02
VAX2	0.37 ± 0.27	0.38 ± 0.22	0.55 ± 0.35	0.35 ± 0.02	0.34 ± 0.02	0.35 ± 0.02
NR1H4	-1.17 ± 0.16	-1.13 ± 0.09	-1.14 ± 0.11	0.17 ± 0.02	0.17 ± 0.01	0.17 ± 0.01
ZNF655	0.56 ± 0.31	0.59 ± 0.34	0.52 ± 0.24	0.33 ± 0.01	0.33 ± 0.01	0.34 ± 0.01
WT1	-1.48 ± 0.19	-1.57 ± 0.11	-1.51 ± 0.14	0.14 ± 0.01	0.13 ± 0.01	0.14 ± 0.01
NKX2-5	-0.97 ± 0.14	-1.00 ± 0.18	-1.00 ± 0.15	0.18 ± 0.01	0.17 ± 0.01	0.18 ± 0.01
GFI1	-0.59 ± 0.23	-0.60 ± 0.17	-0.59 ± 0.21	0.21 ± 0.01	0.21 ± 0.01	0.21 ± 0.01
HOXD13	-0.34 ± 0.21	-0.34 ± 0.20	-0.27 ± 0.25	0.23 ± 0.01	0.23 ± 0.01	0.24 ± 0.02
KLF1	0.66 ± 0.31	0.58 ± 0.30	0.57 ± 0.24	0.38 ± 0.02	0.39 ± 0.02	0.39 ± 0.01
ESX1	-0.04 ± 0.24	-0.01 ± 0.25	0.02 ± 0.21	0.29 ± 0.02	0.29 ± 0.02	0.30 ± 0.02
PAX4	-0.69 ± 0.16	-0.61 ± 0.22	-0.69 ± 0.20	0.23 ± 0.02	0.24 ± 0.02	0.23 ± 0.02
PAX6	0.23 ± 0.09	0.24 ± 0.10	0.24 ± 0.08	0.39 ± 0.01	0.39 ± 0.01	0.39 ± 0.01
SNAI2	-0.49 ± 0.18	-0.45 ± 0.17	-0.38 ± 0.20	0.23 ± 0.01	0.24 ± 0.01	0.24 ± 0.01
GFI1B	-0.63 ± 0.13	-0.54 ± 0.20	-0.59 ± 0.22	0.21 ± 0.01	0.22 ± 0.01	0.21 ± 0.01
PBX4	-0.81 ± 0.12	-0.77 ± 0.15	-0.75 ± 0.14	0.21 ± 0.01	0.21 ± 0.01	0.21 ± 0.02
ZNF200	-0.01 ± 0.12	0.02 ± 0.12	0.01 ± 0.12	0.33 ± 0.01	0.33 ± 0.01	0.33 ± 0.02
POU3F4	-0.88 ± 0.16	-0.82 ± 0.11	-0.89 ± 0.19	0.20 ± 0.01	0.20 ± 0.01	0.20 ± 0.01
FOXC1	-0.76 ± 0.13	-0.75 ± 0.14	-0.69 ± 0.15	0.22 ± 0.02	0.22 ± 0.02	0.23 ± 0.02
NR2E3	0.14 ± 0.15	0.11 ± 0.15	0.10 ± 0.17	0.32 ± 0.02	0.32 ± 0.01	0.32 ± 0.01
ARX	-0.34 ± 0.25	-0.20 ± 0.20	-0.16 ± 0.20	0.27 ± 0.03	0.28 ± 0.02	0.28 ± 0.02
PITX2	0.20 ± 0.21	0.13 ± 0.28	0.24 ± 0.18	0.30 ± 0.01	0.30 ± 0.02	0.30 ± 0.01
CRX	-0.25 ± 0.22	-0.17 ± 0.28	-0.14 ± 0.28	0.22 ± 0.01	0.22 ± 0.01	0.23 ± 0.01
EGR2	1.13 ± 0.26	1.04 ± 0.19	1.06 ± 0.18	0.47 ± 0.01	0.47 ± 0.01	0.47 ± 0.01
HESX1	-0.34 ± 0.19	-0.29 ± 0.22	-0.24 ± 0.17	0.26 ± 0.01	0.26 ± 0.02	0.27 ± 0.01
OVOL2	0.87 ± 0.25	0.82 ± 0.26	0.87 ± 0.27	0.36 ± 0.01	0.36 ± 0.01	0.36 ± 0.01
VENTX	-0.30 ± 0.23	-0.09 ± 0.25	-0.00 ± 0.17	0.29 ± 0.03	0.30 ± 0.03	0.31 ± 0.02
PAX3	-0.22 ± 0.27	-0.17 ± 0.25	-0.08 ± 0.26	0.26 ± 0.02	0.27 ± 0.02	0.28 ± 0.02
POU4F3	-0.47 ± 0.16	-0.50 ± 0.14	-0.45 ± 0.13	0.23 ± 0.01	0.23 ± 0.01	0.24 ± 0.01
VSX1	0.06 ± 0.24	0.08 ± 0.20	0.22 ± 0.19	0.29 ± 0.02	0.29 ± 0.02	0.30 ± 0.02
MSX2	0.24 ± 0.23	0.21 ± 0.23	0.30 ± 0.26	0.32 ± 0.02	0.32 ± 0.01	0.32 ± 0.02
KLF11	-0.20 ± 0.14	-0.08 ± 0.16	-0.02 ± 0.11	0.30 ± 0.02	0.30 ± 0.02	0.32 ± 0.02
PROP1	-0.21 ± 0.24	-0.30 ± 0.22	-0.22 ± 0.23	0.29 ± 0.02	0.28 ± 0.03	0.29 ± 0.02
BCL6	-1.10 ± 0.11	-1.18 ± 0.13	-1.09 ± 0.14	0.18 ± 0.02	0.17 ± 0.01	0.18 ± 0.01
SIX6	1.10 ± 0.31	1.13 ± 0.24	1.24 ± 0.24	0.39 ± 0.02	0.40 ± 0.02	0.41 ± 0.01
PAX7	-0.22 ± 0.16	-0.23 ± 0.19	-0.18 ± 0.15	0.29 ± 0.02	0.29 ± 0.02	0.29 ± 0.01

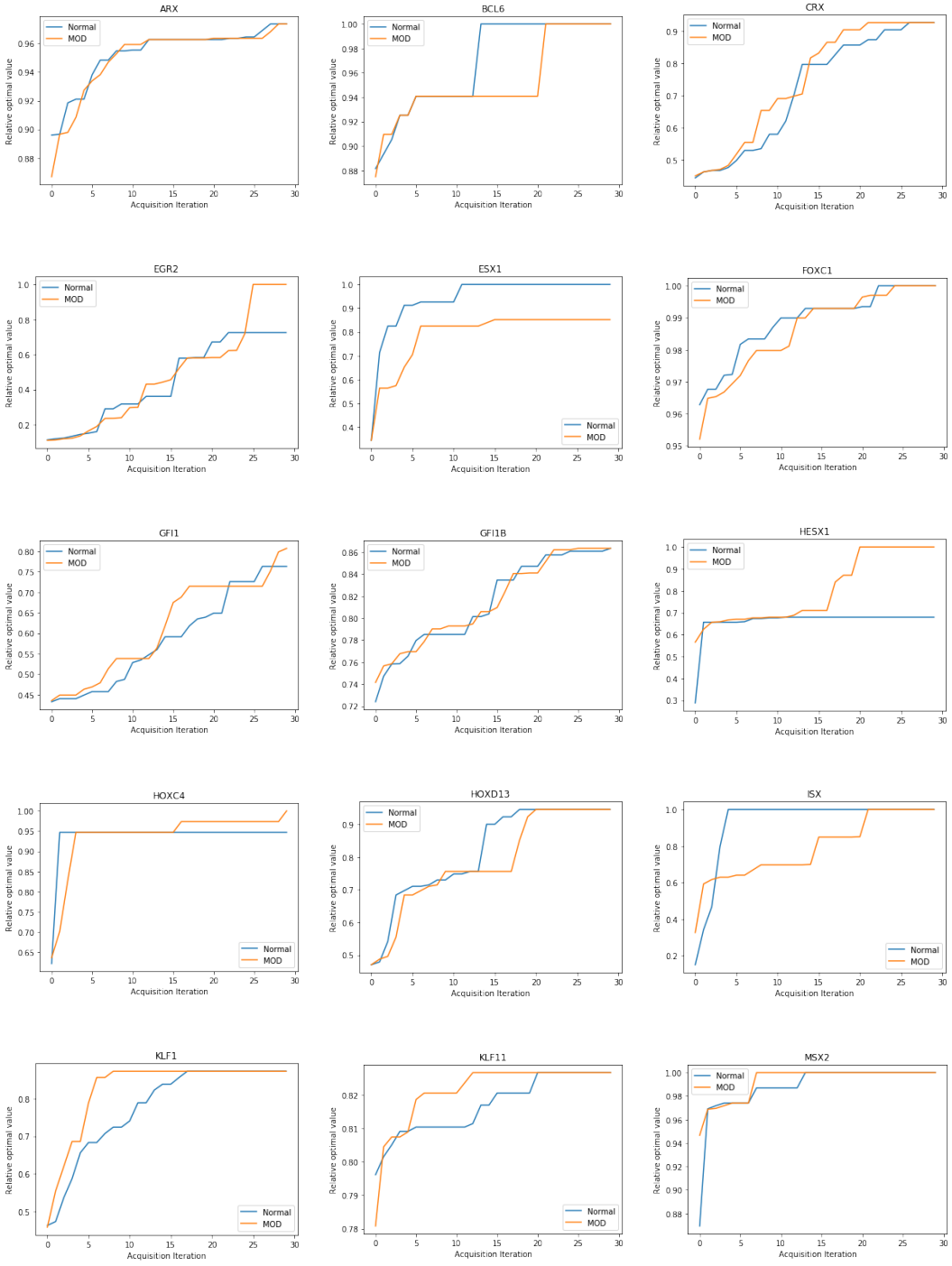
Table 3: NLL and RMSE on in-distribution test set for all 38 TF datasets.

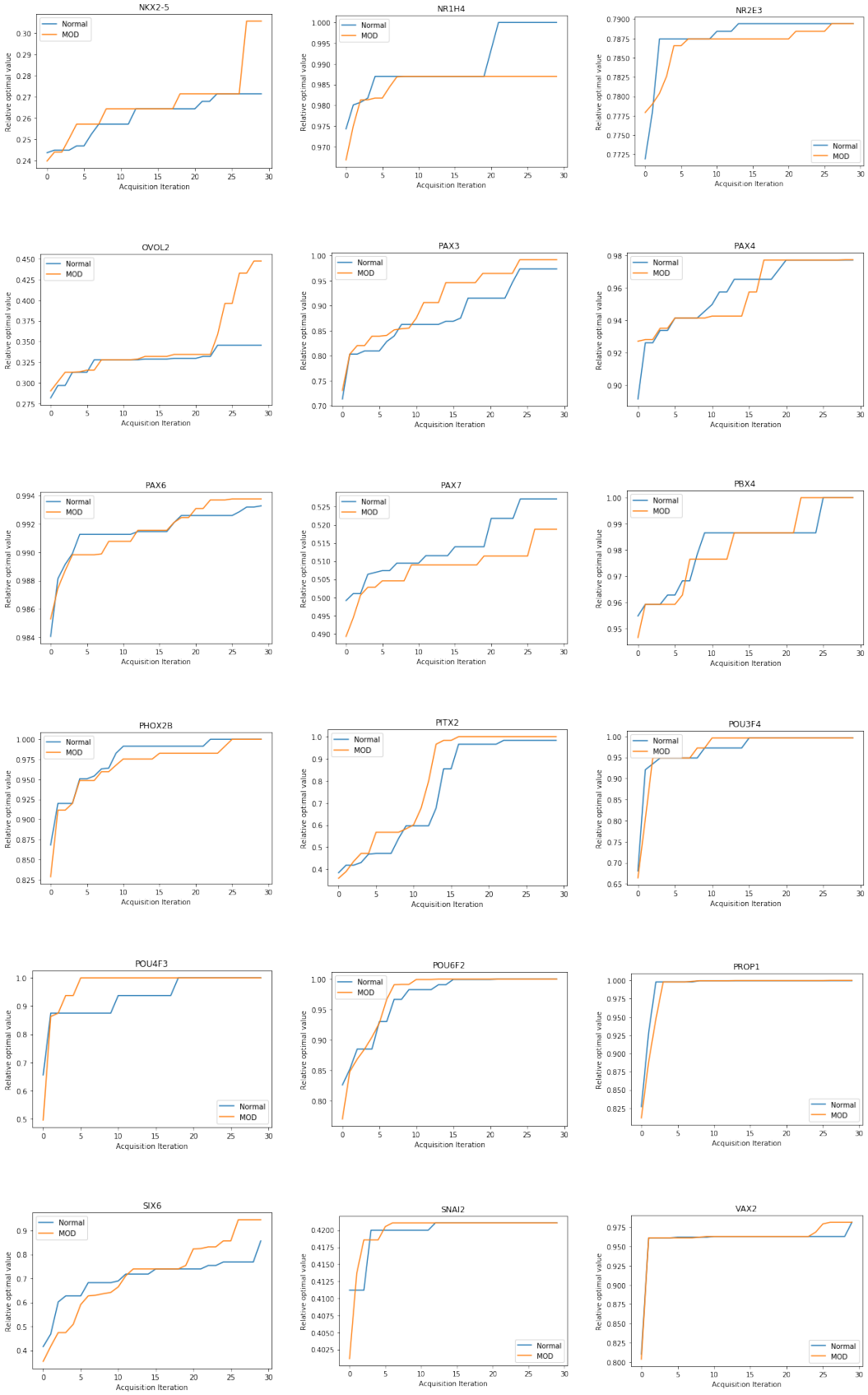
	In-distribution NLL			In-distribution RMSE		
TF	MOD	MTD	Normal	MOD	MTD	Normal
PHOX2B	-1.13 ± 0.04	-1.13 ± 0.05	-1.11 ± 0.04	0.19 ± 0.01	0.19 ± 0.01	0.20 ± 0.01
POU6F2	-1.25 ± 0.04	-1.25 ± 0.04	-1.25 ± 0.04	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01
HOXC4	-1.46 ± 0.05	-1.46 ± 0.04	-1.45 ± 0.05	0.14 ± 0.01	0.14 ± 0.01	0.14 ± 0.01
ISX	-1.39 ± 0.04	-1.39 ± 0.03	-1.39 ± 0.04	0.15 ± 0.01	0.15 ± 0.01	0.15 ± 0.01
VAX2	-1.37 ± 0.05	-1.36 ± 0.05	-1.36 ± 0.04	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
NR1H4	-1.53 ± 0.03	-1.53 ± 0.03	-1.53 ± 0.02	0.13 ± 0.00	0.13 ± 0.00	0.13 ± 0.00
ZNF655	-1.30 ± 0.03	-1.30 ± 0.03	-1.30 ± 0.03	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
WT1	-1.64 ± 0.03	-1.64 ± 0.03	-1.64 ± 0.04	0.12 ± 0.00	0.12 ± 0.00	0.12 ± 0.00
NKX2-5	-1.58 ± 0.05	-1.58 ± 0.03	-1.58 ± 0.03	0.12 ± 0.00	0.12 ± 0.00	0.12 ± 0.00
GFI1	-1.53 ± 0.03	-1.53 ± 0.04	-1.53 ± 0.03	0.13 ± 0.00	0.13 ± 0.00	0.13 ± 0.00
HOXD13	-1.56 ± 0.03	-1.56 ± 0.02	-1.55 ± 0.03	0.13 ± 0.00	0.13 ± 0.00	0.13 ± 0.00
KLF1	-1.11 ± 0.04	-1.11 ± 0.04	-1.11 ± 0.04	0.20 ± 0.01	0.20 ± 0.01	0.20 ± 0.01
ESX1	-1.30 ± 0.04	-1.30 ± 0.04	-1.30 ± 0.04	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
PAX4	-1.32 ± 0.04	-1.32 ± 0.03	-1.32 ± 0.03	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
PAX6	-1.07 ± 0.03	-1.07 ± 0.03	-1.07 ± 0.03	0.21 ± 0.01	0.21 ± 0.01	0.21 ± 0.01
SNAI2	-1.45 ± 0.03	-1.45 ± 0.03	-1.44 ± 0.03	0.14 ± 0.01	0.14 ± 0.00	0.14 ± 0.00
GFI1B	-1.50 ± 0.04	-1.50 ± 0.03	-1.49 ± 0.03	0.14 ± 0.00	0.14 ± 0.00	0.14 ± 0.00
PBX4	-1.41 ± 0.03	-1.41 ± 0.03	-1.41 ± 0.03	0.15 ± 0.00	0.15 ± 0.00	0.15 ± 0.00
ZNF200	-1.16 ± 0.03	-1.16 ± 0.03	-1.16 ± 0.04	0.19 ± 0.01	0.19 ± 0.01	0.19 ± 0.01
POU3F4	-1.50 ± 0.04	-1.50 ± 0.03	-1.49 ± 0.04	0.13 ± 0.00	0.13 ± 0.00	0.14 ± 0.00
FOXC1	-1.33 ± 0.04	-1.34 ± 0.04	-1.33 ± 0.04	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
NR2E3	-1.28 ± 0.03	-1.28 ± 0.03	-1.27 ± 0.03	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01
ARX	-1.25 ± 0.04	-1.25 ± 0.04	-1.24 ± 0.04	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01
PITX2	-1.35 ± 0.04	-1.36 ± 0.04	-1.35 ± 0.04	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
CRX	-1.58 ± 0.04	-1.58 ± 0.04	-1.57 ± 0.04	0.12 ± 0.01	0.12 ± 0.01	0.13 ± 0.00
EGR2	-1.06 ± 0.04	-1.06 ± 0.03	-1.06 ± 0.03	0.21 ± 0.01	0.21 ± 0.01	0.21 ± 0.01
HESX1	-1.34 ± 0.03	-1.33 ± 0.03	-1.33 ± 0.03	0.16 ± 0.00	0.16 ± 0.00	0.16 ± 0.00
OVOL2	-1.33 ± 0.03	-1.33 ± 0.03	-1.32 ± 0.04	0.16 ± 0.00	0.16 ± 0.00	0.16 ± 0.01
VENTX	-1.17 ± 0.05	-1.17 ± 0.06	-1.16 ± 0.06	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01
PAX3	-1.32 ± 0.04	-1.33 ± 0.03	-1.31 ± 0.03	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
POU4F3	-1.50 ± 0.03	-1.50 ± 0.03	-1.50 ± 0.04	0.13 ± 0.00	0.13 ± 0.00	0.14 ± 0.00
VSX1	-1.38 ± 0.04	-1.38 ± 0.04	-1.37 ± 0.04	0.15 ± 0.01	0.15 ± 0.01	0.15 ± 0.01
MSX2	-1.37 ± 0.04	-1.37 ± 0.04	-1.37 ± 0.04	0.15 ± 0.01	0.15 ± 0.01	0.16 ± 0.01
KLF11	-1.20 ± 0.04	-1.19 ± 0.03	-1.19 ± 0.03	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01
PROP1	-1.32 ± 0.04	-1.32 ± 0.04	-1.33 ± 0.03	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.01
BCL6	-1.54 ± 0.03	-1.55 ± 0.03	-1.54 ± 0.04	0.13 ± 0.00	0.13 ± 0.00	0.13 ± 0.00
SIX6	-1.22 ± 0.04	-1.22 ± 0.04	-1.22 ± 0.04	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01
PAX7	-1.27 ± 0.04	-1.27 ± 0.04	-1.27 ± 0.04	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01

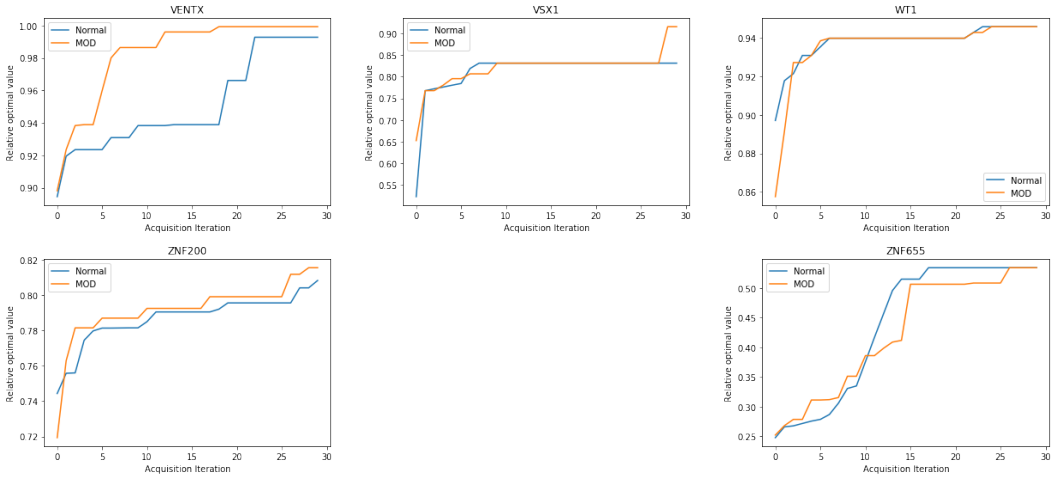
Table 4: Mean fraction (across 10 BO runs) of top 1% of points retrieved.

TF	MOD	Normal
ARX	**0.191463**	0.184451
BCL6	0.400610	**0.403354**
CRX	**0.187805**	0.182012
EGR2	**0.213720**	0.153659
ESX1	0.464634	**0.492378**
FOXC1	0.198781	**0.212195**
GFI1B	**0.089634**	0.089329
GFI1	**0.145427**	0.114634
HESX1	**0.433232**	0.403354
HOXC4	**0.449695**	0.441463
HOXD13	**0.402439**	0.402134
ISX	0.401829	**0.426524**
KLF11	**0.295427**	0.271037
KLF1	**0.327744**	0.304573
MSX2	**0.507622**	0.492378
NKX2-5	0.328049	**0.329573**
NR1H4	**0.242073**	0.236890
NR2E3	**0.102744**	0.100000
OVOL2	**0.029878**	0.002134
PAX3	0.304878	**0.317378**
PAX4	0.308232	**0.327134**
PAX6	**0.018902**	0.017683
PAX7	0.224695	**0.234756**
PBX4	**0.262805**	0.249695
PHOX2B	**0.355793**	0.333537
PITX2	**0.258537**	0.228354
POU3F4	0.512805	**0.525305**
POU4F3	**0.506097**	0.496646
POU6F2	0.360061	**0.369512**
PROP1	**0.490854**	0.427439
SIX6	**0.261585**	0.253963
SNAI2	**0.224390**	0.221341
VAX2	**0.443293**	0.428354
VENTX	**0.214024**	0.201220
VSX1	**0.483232**	0.466463
WT1	0.292683	**0.310061**
ZNF200	0.084451	**0.100915**
ZNF655	**0.107012**	0.086585

Figure 2: Bayesian optimization performance of MOD ensemble vs. Normal ensemble. The relative optimal value r_T values at each acquisition iteration are averaged over 10 replicate BO runs.




NKX2-5
NR1H4
NR2E3
OVOL2
PAX3
PAX4
PAX6
PAX7
PBX4
PHOX2B
PITX2
POU3F4
POU4F3
POU6F2
PROP1
SIX6
SNAI2
VAX2
Normal
MOD
Relative optimal value
Acquisition Iteration


VENTX
VSX1
WT1
ZNF200
ZNF655
Normal
MOD
Relative optimal value
Acquisition Iteration

References

[1] Breiman L (1996) Bagging predictors. *Machine Learning* 24: 123-140.

[2] Brown G (2004) Diversity in neural network ensembles. Ph.D. thesis, University of Birmingham.

[3] Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Advances in Neural Information Processing Systems*.

[4] Osband I, Blundell C, Pritzel A, Van Roy B (2016) Deep exploration via bootstrapped DQN. In: *Advances in Neural Information Processing Systems*.

[5] Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* 51: 181–207.

[6] Papadopoulos G, Edwards PJ, Murray AF (2001) Confidence estimation methods for neural networks: A practical comparison. *IEEE Transactions on Neural Networks* 12: 1278-1287.

[7] Balan AK, Rathod V, Murphy KP, Welling M (2015) Bayesian dark knowledge. In: *Advances in Neural Information Processing Systems*.

[8] Beluch WH, Genewein T, Nürnberger A, Köhler JM (2018) The power of ensembles for active learning in image classification. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

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

[10] Chen RY, Sidor S, Abbeel P, Schulman J (2017) UCB exploration via Q-ensembles. *arXiv:170601502* .

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

[12] Hafner D, Tran D, Lillicrap T, Irpan A, Davidson J (2018) Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *arXiv:180709289* .

[13] Lee K, Lee H, Lee K, Shin J (2018) Training confidence-calibrated classifiers for detecting out-of-distribution samples. In: *International Conference on Learning Representations*.

[14] Lee S, Purushwalkam S, Cogswell M, Crandall D, Batra D (2015) Why M heads are better than one: Training a diverse ensemble of deep networks. *arXiv:151106314* .

[15] Barrera LA, Vedenko A, Kurland JV, Rogers JM, Gisselbrecht SS, et al. (2016) Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science* 351: 1450–1454.

[16] Hashimoto TB, Yadlowsky S, Duchi JC (2018) Derivative free optimization via repeated classification. In: *International Conference on Artificial Intelligence and Statistics*.

[17] Hernández-Lobato JM, Requeima J, Pyzer-Knapp EO, Aspuru-Guzik A (2017) Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. *arXiv:170601825* .