A Notation

Table 2: The notations used in the paper.

Notation	Description
$\boldsymbol{x}$	the input vector
$\{\boldsymbol{x}_i\}_{i\in I}$	a set of input vectors indexed by i
f	the underlying function
$\mathrm{y}_{\boldsymbol{x}}$	the observed value of the function at a given location $\boldsymbol{x}$
$\sigma_n(\boldsymbol{x})$	the standard variance of the observation noise at $\boldsymbol{x}$
σ_n	the standard derivation of a homoscedastic observation noise
$\sigma_{\mathbf{x}}$	standard derivation of $f(\mathbf{x})$
η	the prior variance for BNNs

B The Pseudocodes

Algorithm 1 A procedure of (Transductive) Active Learning. We use red and blue to show the difference between active learning and TAL. TIG and MIG can be replaced by any other acquisition functions.

Require: *Selection Model*: $\mathcal{M}^s$; *Prediction Model*: $\mathcal{M}^p$.
Require: Datasets: $\mathcal{D}_{\text{tr}} = \{\boldsymbol{X}_{\text{tr}}, \boldsymbol{y}_{\text{tr}}\}$, $\mathcal{D}_{\text{te}} = \{\boldsymbol{X}_{\text{te}}, \boldsymbol{y}_{\text{te}}\}$, $\mathcal{D}_{\text{pl}} = \{\boldsymbol{X}_{\text{pl}}, \boldsymbol{y}_{\text{pl}}\}$.
Require: Total active learning iterations: T; #Queried samples per iteration: m.
$\mathcal{R} = \emptyset$.
for $t = 1$ to T **do**
  Train $\mathcal{M}^p, \mathcal{M}^s$ on $\mathcal{D}_{\text{tr}}$ until convergence.
  Test $\mathcal{M}^p$ over $\mathcal{D}_{\text{te}}$ and put the result to $\mathcal{R}$.
  InfoG = TIG($\boldsymbol{X}_{\text{pl}}, \mathcal{M}^s$) or InfoG = MIG($\boldsymbol{X}_{\text{pl}}, \boldsymbol{X}_{\text{te}}, \mathcal{M}^s$).
  Sort InfoG in descending order and retrieve top m samples from $\mathcal{D}_{\text{pl}}$ as $\mathcal{D}_{\text{qe}}$.
  $\mathcal{D}_{\text{tr}} \leftarrow \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{qe}}$; $\mathcal{D}_{\text{pl}} \leftarrow \mathcal{D}_{\text{pl}} \setminus \mathcal{D}_{\text{qe}}$.
  $t \leftarrow t + 1$.
end for
return $\mathcal{R}, \mathcal{M}^p, \mathcal{D}_{\text{tr}}$.

C Information Gains for Active Learning

We introduce three types of information gains and present their analytical forms for Gaussian predictive distributions. Then, we provide a greedy approximation for computing the optimal batch corresponding to BatchMIG.

C.1 Three Types of Information Gains

We firstly specify the analytic expressions for computing the information gain acquisition functions:

Total Information Gain (TIG), measures the mutual information between the queried point $\mathbf{x}$ and the model parameters $\mathbf{w}$,

$$\mathrm{TIG}(\boldsymbol{x}) \coloneqq \mathbb{I}(y_{\mathbf{x}}; \mathbf{w} | \mathcal{D}_{\text{tr}}) \overset{\text{Gaussian predictive dist}}{=} \frac{1}{2} \log\left(1 + \sigma^2_{\boldsymbol{x}} / \sigma_n(\mathbf{x})^2\right), \tag{4}$$

Marginal Information Gain (MIG), measures the mutual information between the queried point $\mathbf{x}$ and a point $\mathbf{x}_u$ of interest,

$$\mathrm{MIG}(\boldsymbol{x}; \boldsymbol{x}_u) \coloneqq \mathbb{I}(y_{\mathbf{x}}; f(\mathbf{x}_u) | \mathcal{D}_{\text{tr}}) \overset{\text{Gaussian predictive dist}}{=} -\frac{1}{2} \log\left(1 - \frac{\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{x}_u)^2}{\sigma^2_{\boldsymbol{x}_u}(\sigma^2_{\boldsymbol{x}} + \sigma_n(\boldsymbol{x})^2}\right), \tag{5}$$

Algorithm 2 Computing XLL and XLLR.

Require: Model Predictions $\{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}_{i=1}^m$; Test set $\mathcal{D}_{\text{te}}$; Batch size b

for $j = 1$ to m **do** ▷ Reference Model
  for $i = 1$ to m **do** ▷ Normalize Predictive Marginals
    $\boldsymbol{D}_i^0 = \sqrt{\text{diag}(\boldsymbol{\Sigma}_j)}/\sqrt{\text{diag}(\boldsymbol{\Sigma}_i)}$.
    $\bar{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_j, \bar{\boldsymbol{\Sigma}}_i = \boldsymbol{D}_i^0 \boldsymbol{\Sigma}_i \boldsymbol{D}_i^0$.
  end for
  $\mathcal{T}' = \{\}$.
  for $(\mathbf{x}, y) \in \mathcal{D}_{\text{te}}$ **do** ▷ Build Test Batches
    Top correlated points $\mathcal{B}_{\mathbf{x}} := \{(\mathbf{x}_k, y_k)\}_{k=1}^b$; Add $\mathcal{B}_{\mathbf{x}}$ to $\mathcal{T}'$.
  end for
  for $i = 1$ to m **do** ▷ Compute Log Joints
    $\text{lld}_i^j = \frac{1}{|\mathcal{T}'|} \sum_{\mathcal{B} \in \mathcal{T}'} \log \mathcal{N}(\mathcal{B} | \bar{\boldsymbol{\mu}}_i, \bar{\boldsymbol{\Sigma}}_i)$.
  end for
  $\{\text{rank}_i^j\}_{i=1}^m$ from sorting $\{\text{lld}_i^j\}_{i=1}^m$.
end for
$\text{lld}_i = \frac{1}{m} \sum_{j=1}^m \text{lld}_i^j$, $\text{rank}_i = \frac{1}{m} \sum_{j=1}^m \text{rank}_i^j$ ▷ Average over References
return $\{\text{lld}_i\}_{i=1}^m$ and $\{\text{rank}_i\}_{i=1}^m$.

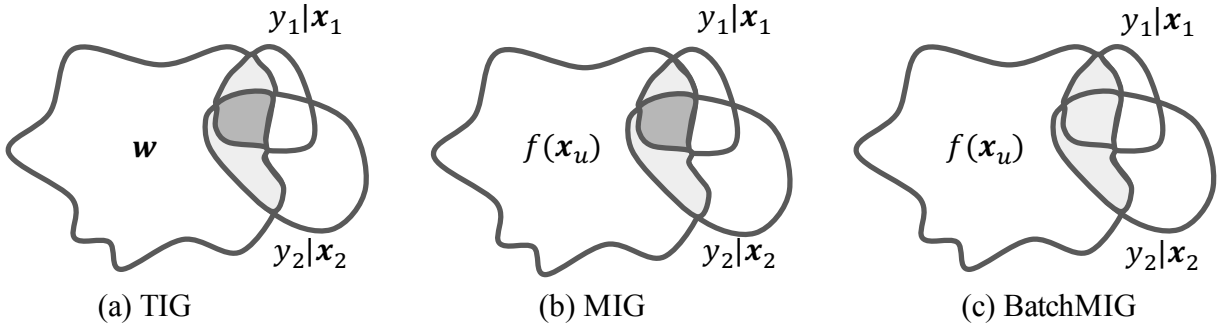

Figure 8: An illustration of how do TIG, MIG and BatchMIG compute the informativeness of two candidate points. TIG measures the mutual information between data and model, whereas MIG and BatchMIG measure that between data and test points. Dark regions represent the information gain is double-counted, *i.e.,* both TIG and MIG overestimate the gain.

Batched Marginal Information Gain (BatchMIG), measures the mutual information between a batch of queried points $\mathbf{x}_{1:q}$ and the point $\mathbf{x}_u$ of interest,

$$\begin{aligned}
&\text{BatchMIG}(\boldsymbol{x}_{1:q}; \boldsymbol{x}_u) \coloneqq \mathbb{I}(y_{\mathbf{x}_{1:q}}; f(\mathbf{x}_u) | \mathcal{D}_{\text{tr}}) \\
&\overset{\text{Gaussian predictive dist}}{=} -\frac{1}{2} \log \left(1 - \frac{\text{Cov}\left(\boldsymbol{x}_{1:q}, \boldsymbol{x}_u\right)^\top \left(\text{Cov}\left(\boldsymbol{x}_{1:q}, \boldsymbol{x}_{1:q}\right) + \boldsymbol{\sigma}_n^2(\boldsymbol{x}_{1:q})\right)^{-1} \text{Cov}\left(\boldsymbol{x}_{1:q}, \boldsymbol{x}_u\right)}{\sigma_{\boldsymbol{x}_u}^2} \right),
\end{aligned} \tag{6}$$

Again for MIG and BatchMIG, assuming that we are interested at a set of points $\{\mathbf{x}_u^i\}_{i=1}^I$, as recommended in MacKay (1992), we adopt the mean marginal information gains: $\frac{1}{I} \sum_{i=1}^I \text{MIG}(\boldsymbol{x}; \boldsymbol{x}_u^i)$ and $\frac{1}{I} \sum_{i=1}^I \text{BatchMIG}(\boldsymbol{x}; \boldsymbol{x}_u^i)$.

C.2 A Greedy Approximation of the Optimal Batch

In practice we will usually query a batch of points at each iteration for efficiency. For TIG and MIG, selecting a batch corresponds to selecting the points with highest information gains, correspondingly. For BatchMIG, although extending the information gain acquisition functions from the single-point scenario to the batch scenario is straightforward, solving for the optimal batch requires a combinatorial explosion of computations. Following (Kirsch et al., 2019), we adopt a greedy approximation of the optimal batch, which is specified in Alg 3.

Algorithm 3 BatchBald (Kirsch et al., 2019): a greedy approximation of the optimal batch.

Require: Model $\mathcal{M}$, Points of interest $\mathcal{I}$, Query Batch Size q
Require: The information gain acquisition function IG.
1: $A \leftarrow \emptyset$
2: **for** $i = 1$ to q **do**
3: $\quad \mathbf{x}^\star \in \arg\max_{\mathbf{x}} \mathrm{IG}(\{\mathbf{x}\} \cup A, \mathcal{M}, \mathcal{I})$
4: $\quad A = A \cup \{\mathbf{x}^\star\}$
5: **end for**
6: **return** A.

D Experimental Details

D.1 Hyperparameters

We use the standard regression task for tuning hyperparameters with respect to each method and each dataset. Specifically, we split the dataset into train (60%), valid (20%) and test (20%). Across 10 different runs, we use the same validation set but split train/test randomly. Finally the averaged validation log likelihood will be used for selecting the hyperparameters. A list of details about hyperparameters is shown in Table 3.

With the tuned hyperparameters, we conduct transductive active learning and compute the XN-LLDR metrics. To avoid that the test set being used for tuning hyper-parameters, we make sure the randomly selected test set is disjoint with the validation set for hyperparameter tuning.

Table 3: The hyperparameters for each method

Methods	Hyperparameters to tune
BBB	lr: $[0.001, 0.003, 0.01]$, hidden units: $[50, 400]$, #eval_cov_samples: $[100, 700, 5000]$
NNG	lr: $[0.001, 0.003, 0.01]$, hidden units: $[50, 400]$
HMC	lr: $[0.001, 0.003, 0.01]$, hidden units: $[50, 400]$
FBNN	lr: $[0.001, 0.003, 0.01]$, number of random measurement points $[5, 20, 100]$, hidden units: $[50, 400]$
Dropout	lr: $[0.001, 0.003, 0.01]$, hidden units: $[50, 400]$, Dropout Rate: $[0.0025, 0.01, 0.05]$, Observation variance: $[0.005, 0.025, 0.125]$
Ensemble	lr: $[0.001, 0.003, 0.01]$, hidden units: $[50, 400]$
Methods	Other Settings
(SV)GP	Optimizer=Adam, lr=0.003, epochs=10,000, batch_size=min(5,000, #training data), lenghth_scale are initialized with k-means on training data, ARD=True, min_obsvar=1e-5 (except for `Wine` dataset, we use min_obsvar = 1e-8); For large datasets, we adopt SVGP with 1,000 inducing points; For (SV)GP-NKN, we adopt the same NKN as in Sun et al. (2018) and epochs=5,000.
BBB	Optimizer=Adam, epochs=10,000, batch_size=100, #training_particles=10, #test_particles=5,000.
NNG	Optimizer=NG-KFAC(damping=1e-5, ema_cov_decay=0.999), epochs=10,000, lr decay by a factor 0.1 every 5000 epochs, #training_particles=10, #test_particles=5,000, #eval_cov_samples=5000.
HMC	#chains = 10, burnin=5,000 for small datasets and 15,000 for larger ones, step_size starts at 0.01 but is adapted according to the acceptance rate, #leap_frog_steps=5; We select one particle every 100 samples after burnin untill we collected 100 samples in each chain, which results at 1,000 samples for testing and computing the covariance. We use Adam Optimizer for optimizing the prior hyperparameters η, ξ every 10 HMC steps.
FBNN	Optimizer=Adam, epochs=10,000, batch_size=#training data for small datasets and 900 for larger datasets in order to match the computation complexity of SVGP. The network has 400 hidden units with cosine activations.
Dropout	Optimizer=Adam, epochs=10,000, batch_size=100. We use $5,000$ samples for test and computing the covariance. L2 regularization with $10^{-4} * (1 - \text{dropout\_rate})/(2. * N * \xi)$.
Ensemble	Optimizer=Adam, epochs=10,000, batch_size=100, #networks=100.

E Additional Results

We present here the additional results, including (1) Log Joints versus Log Marginals; (2) Average Rank in TAL; (3) Average Log Joint Likelihood on UCI datasets; (4) RMSE performance of TAL using different Acquisition functions; (5) Comparisons between Different Data Acquisition Functions; (6) TAL Results of Different Models on Synthetic Dataset.

E.1 Log Joints versus Log Marginals

We visualize the scatter plot of the joint log-likelihoods and the marginal log-likelihoods in Figure 9. We observe that the joint log-likelihood is positively correlated with the marginal log-likelihood.

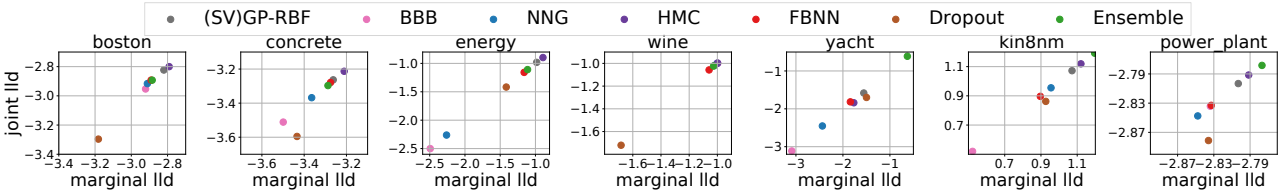

Figure 9: Scatters of log joint likelihoods versus log marginal likelihoods where each point represents one method. The log joints are computed over random batches with 5 points.

E.2 Average Rank in TAL (Table 4)

Table 4: Average rank of each method's LLD and RMSE on TAL at the last iteration with different prediction models. We use red to highlight the best ones, and blue for the worst ones.

	Prediction Model/Selection Model	(SV)GP-RBF	BBB	NNG	HMC	FBNN	Dropout	Ensemble
RMSE	Oracle	2.4	4.3	3.8	2.0	2.4	3.0	3.0
	Dropout	2.8	4.4	4.3	1.4	2.4	2.5	3.2
	(SV)GP-RBF	2.4	4.3	3.7	1.6	2.0	3.2	3.8
	NNG	2.7	3.9	3.7	1.9	2.1	3.2	3.5
	HMC	2.4	4.4	3.4	2.5	1.8	3.4	3.1
	Average Rank	2.5	4.3	3.8	1.9	2.1	3.1	3.3
LLD	Oracle	2.1	4.5	4.0	1.8	2.2	4.0	2.5
	Dropout	2.8	4.5	4.1	1.8	2.6	2.6	2.6
	(SV)GP-RBF	2.5	4.2	3.8	1.7	2.1	3.3	3.3
	NNG	2.7	3.9	3.8	2.1	2.0	3.3	3.3
	HMC	2.8	4.5	3.2	2.5	2.0	3.9	2.2
	Average Rank	2.6	4.3	3.8	2.0	2.2	3.4	2.8

Table 4 shows the results of mixing and matching a wider variety of training and selection models. In general, we observe that regardless of which model is used for training, the best results are obtained when queries are selected using the most accurate models, rather than the same models used for training. We believe this experiment directly indicates that high-quality posterior predictive distributions are useful for data selection, above and beyond the benefits from making better predictions from a fixed training set.

E.3 Average XLLs on UCI datasets (Table 5)

Table 5: The average XLL for each model on UCI datasets.

Dataset/Method	(SV)GP-RBF	BBB	NNG	HMC	FBNN	Dropout	Ensemble
Boston	-3.217 (0.134)	-3.316 (0.156)	-3.202 (0.133)	-3.177 (0.133)	-3.237 (0.138)	-3.456 (0.160)	-3.202 (0.139)
Concrete	-3.342 (0.015)	-3.394 (0.018)	-3.351 (0.016)	-3.336 (0.015)	-3.344 (0.015)	-3.615 (0.029)	-3.340 (0.015)
Energy	-1.382 (0.065)	-1.430 (0.068)	-1.437 (0.068)	-1.378 (0.064)	-1.384 (0.064)	-1.434 (0.067)	-1.386 (0.065)
Wine	-1.215 (0.032)	-1.266 (0.038)	-1.228 (0.034)	-1.224 (0.034)	-1.222 (0.035)	-1.306 (0.042)	-1.226 (0.034)
Yacht	-2.062 (0.115)	-2.112 (0.108)	-2.074 (0.102)	-2.126 (0.118)	-2.011 (0.103)	-2.674 (0.166)	-1.998 (0.102)
Kin8nm	0.902 (0.031)	0.892 (0.031)	0.890 (0.032)	0.902 (0.031)	0.901 (0.031)	0.796 (0.032)	0.897 (0.031)
Naval	6.853 (0.172)	6.795 (0.176)	6.811 (0.175)	6.882 (0.166)	6.870 (0.171)	6.971 (0.163)	6.920 (0.173)
Power_plant	-2.793 (0.015)	-2.812 (0.018)	-2.821 (0.019)	-2.801 (0.017)	-2.806 (0.017)	-2.828 (0.015)	-2.796 (0.016)

E.4 RMSE performance of TAL using different Acquisition functions.

In addition to the LLD performance, we also present the RMSE performance of TAL using BatchMIG, MIG, TIG and random selection with the 'Oracle' model in Figure 10 (right part).

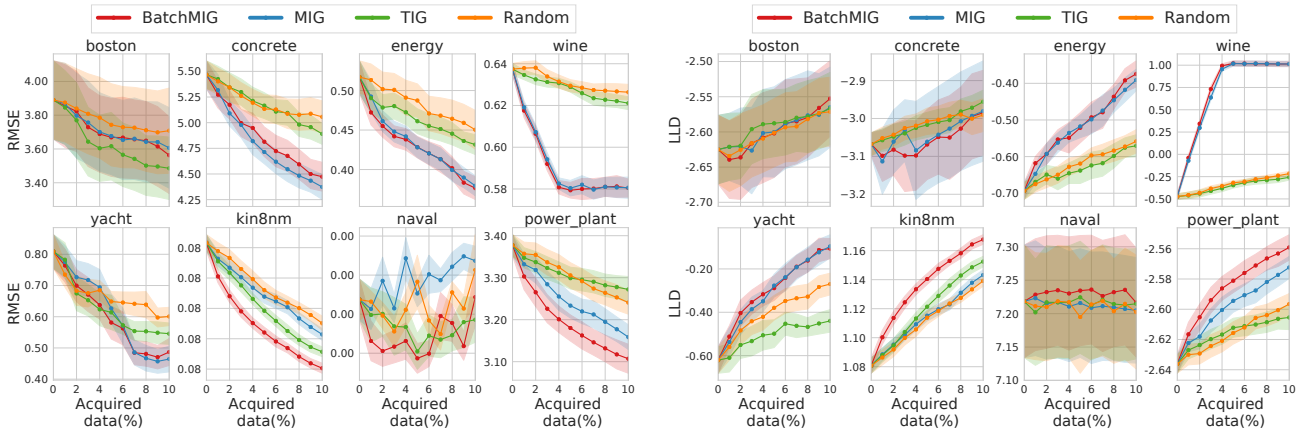

Figure 10: RMSE and LLD performance of TAL with the pre-trained NKN kernel (Oracle).

E.5 Comparisons between Different Data Acquisition Functions

We present here the results using different data acquisition functions on synthetic datasets and on UCI datasets. The results can be found in Figure 11 and Figure 12, where we can observe that TAL acquisition functions consistently outperform other criterions.

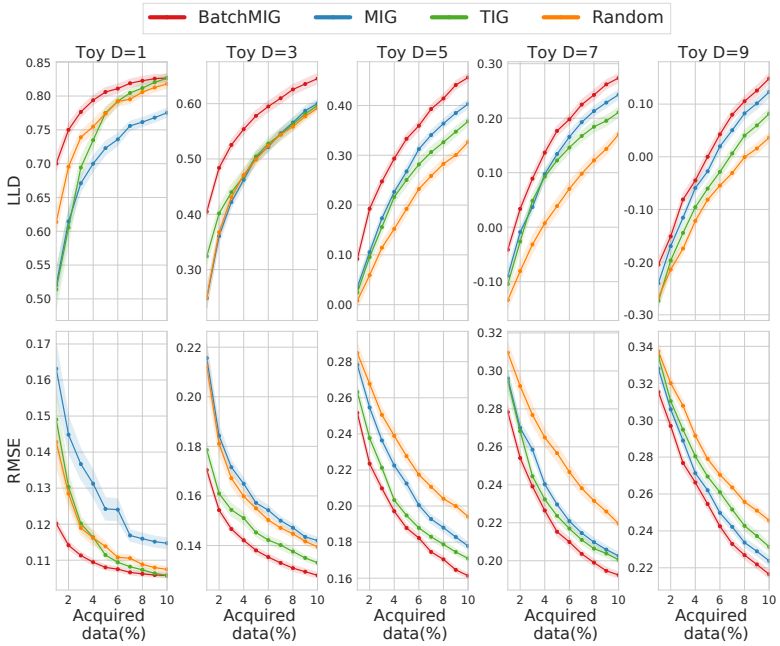

Figure 11: Comparisons between different acquisition functions with Oracle model.

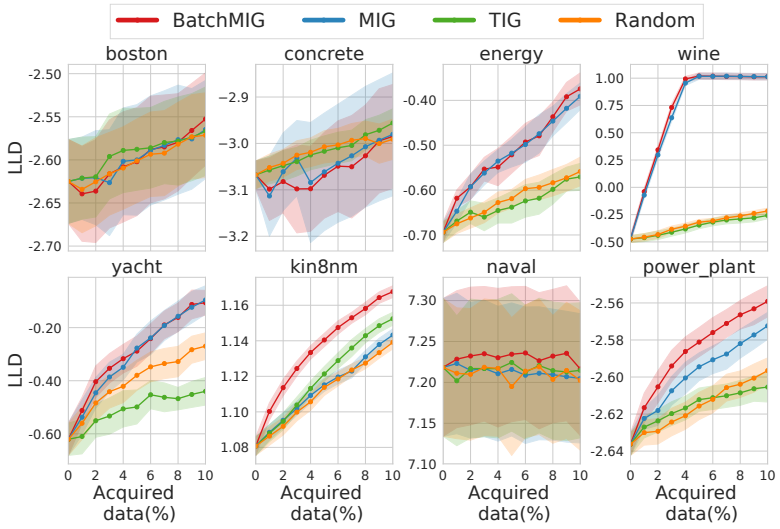

Figure 12: Right: Comparing TAL criteria on UCI datasets using the Oracle (NKN) model.

E.6 TAL Results of Different Models on Synthetic Datasets

To evaluate how each models perform on TAL, we compare them with BatchMIG and TIG on the synthetic datasets. The results are presented in Figure 13 and Figure 14 respectively.

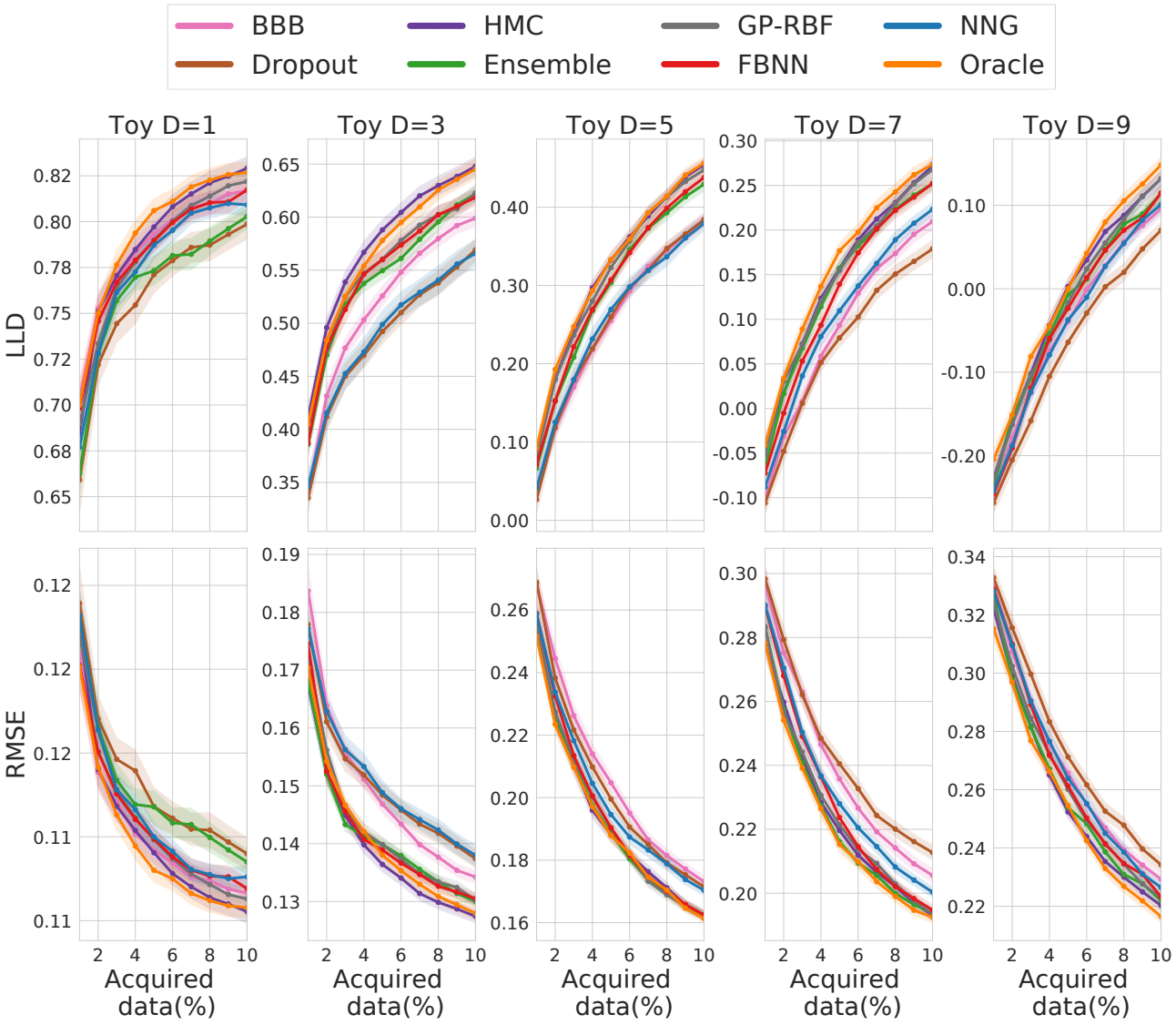

Figure 13: BatchMIG on toy datasets, with fixed observation variance.

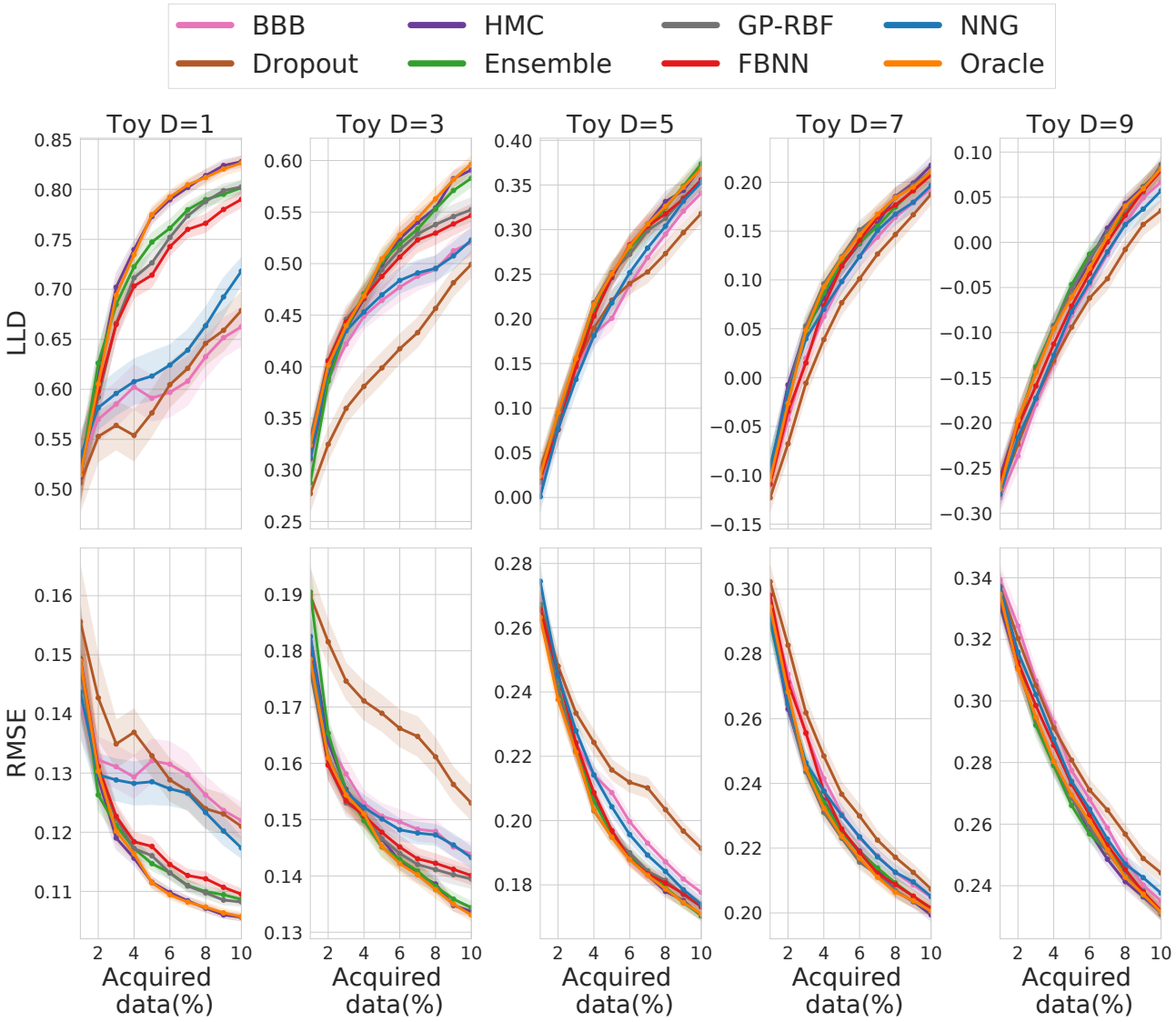

Figure 14: TIG on toy datasets, with fixed observation variance.

F A Theoretical Connection between Log Likelihoods and Predictive Correlations

To understand why XLL directly reflects the accuracy of the correlations, consider the following distributions:

$$
\begin{aligned}
p(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\text{gen}}, \operatorname{diag}(\boldsymbol{\sigma}_{\text{gen}})\boldsymbol{C}_{\text{gen}} \operatorname{diag}(\boldsymbol{\sigma}_{\text{gen}})),\\
q(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\text{ref}}, \operatorname{diag}(\boldsymbol{\sigma}_{\text{ref}})\boldsymbol{C} \operatorname{diag}(\boldsymbol{\sigma}_{\text{ref}})),\\
p_m(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\text{gen}}, \operatorname{diag}(\boldsymbol{\sigma}_{\text{gen}})),\\
q_m(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\text{ref}}, \operatorname{diag}(\boldsymbol{\sigma}_{\text{ref}})),\\
p_c(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{C}_{\text{gen}}),\\
q_c(\boldsymbol{y}|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{C}),
\end{aligned} \tag{7}
$$

where $p(\boldsymbol{y}|\boldsymbol{X})$ is the data generating distribution, and $\boldsymbol{\mu}_{\text{gen}}$, $\boldsymbol{\sigma}^2_{\text{gen}}$ and $\boldsymbol{C}_{\text{gen}}$ are the ground-truth mean, variance and correlations respectively. Observe that $-\mathrm{KL}\left(p\|q\right)$ is the quantity that XLL is approximating using samples (up to a constant), while $\mathrm{KL}\left(p_c\|q_c\right)$ is a measure of dissimilarity between the correlation matrices or the LogDet divergence between two positive semidefinite matrices $\boldsymbol{C}_{\text{gen}}$ and $\boldsymbol{C}$. We now show that, if the reference marginals (i.e., $\boldsymbol{\mu}_{\text{ref}}, \boldsymbol{\sigma}_{\text{ref}}$) are close to the ground truth marginals, then $\mathrm{KL}\left(p\|q\right)$ approximately equals $\mathrm{KL}\left(p_c\|q_c\right)$. Hence, XLL can be seen as a measure of the accuracy of the predictive correlations.

Theorem 1. *Let the predictive distributions be defined above, and let b be the number of points for evaluation, λ denote the smallest eigenvalue of $\mathbf{C}$ and $\xi = \mathrm{KL}\left(p_m\|q_m\right)$. If $\xi \ll 1$, then we have:*

$$
\left|\mathrm{KL}\left(p\|q\right) - \mathrm{KL}\left(p_c\|q_c\right)\right| = \mathcal{O}\left(\frac{b^{3/2}}{\lambda}\sqrt{\xi}\right). \tag{8}
$$

Remark 1. *Because the expected joint log-likelihood $\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{X})} \log q(\boldsymbol{y}|\boldsymbol{X}) = \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{X})} \log p(\boldsymbol{y}|\boldsymbol{X}) - \mathrm{KL}\left(p\|q\right)$, this theorem illustrates that, for nearly-optimal reference marginals, the expected joint log-likelihood reflects the quality of the predictive correlations. This validates the reliability of XLL.*

Remark 2. *In practice, the predictive covariance is $\boldsymbol{\Sigma} + \sigma_n^2\mathbb{I}$, where σ_n^2 is the variance of the modeled observation noise and $\boldsymbol{\Sigma}$ is the predictive covariance for the underlying function. In general, σ_n^2 and the predictive variances in $\boldsymbol{\Sigma}$ are in the same order of magnitude. Therefore, the smallest eigenvalue λ of the correlation matrix $\mathbf{C}$ is not much smaller than 1. Furthermore, b is small because we evaluate XLL and XLLR over mini-batches ($b = 5$ in our experiments).*

As suggested by the theorem, the ideal reference model would be the oracle, i.e. the true data generating distribution. However, in practice we only have access to models which imperfectly match the distribution. Fortunately, we found that the relative order of XLL values do not appear to be overly sensitive to the choice of reference model. Therefore, to further avoid favoring any particular model as the reference, we propose to iterate through every candidate model to act as the reference model once. Then, for each candidate model, we average its XLL or XLLR across all reference models. Empirically, we found that XLL and XLLR align well with the corresponding performance in TAL benchmarks as well as the oracle-based meta-correlations. In below, we provide the proof of Theorem 1:

Proof. We first define:

$$\boldsymbol{d} := \frac{\boldsymbol{\mu}_{\text{gen}} - \boldsymbol{\mu}_{\text{ref}}}{\boldsymbol{\sigma}_{\text{ref}}}, \quad \boldsymbol{r} := \frac{\boldsymbol{\sigma}_{\text{gen}}}{\boldsymbol{\sigma}_{\text{ref}}}, \tag{9}$$

and let $\mathbf{1} \in \mathbb{R}^{b\times b}$ be the all-ones matrix and $\mathbb{I}$ be the identity matrix, then we have:

$$\begin{aligned}
2\mathrm{KL}\left(p\|q\right) =& \log \frac{|\mathrm{diag}(\boldsymbol{\sigma}_{\text{gen}})\boldsymbol{C}\mathrm{diag}(\boldsymbol{\sigma}_{\text{gen}})|}{|\mathrm{diag}(\boldsymbol{\sigma}_{\text{ref}})\boldsymbol{C}_{\text{gen}}\mathrm{diag}(\boldsymbol{\sigma}_{\text{ref}})|} - b + \mathrm{tr}\left(\boldsymbol{C}^{-1}\mathrm{diag}(\boldsymbol{r})\boldsymbol{C}_{\text{gen}}\mathrm{diag}(\boldsymbol{r})\right) + \boldsymbol{d}^\top\boldsymbol{C}^{-1}\boldsymbol{d} \\
=& \sum_{i=1}^{b} \log \frac{\boldsymbol{\sigma}_{\text{ref},i}^2}{\boldsymbol{\sigma}_{\text{gen},i}^2} + \log\frac{|\boldsymbol{C}|}{|\boldsymbol{C}_{\text{gen}}|} - b + \boldsymbol{r}^\top\left(\boldsymbol{C}^{-1}\circ\boldsymbol{C}_{\text{gen}}\right)\boldsymbol{r} + \boldsymbol{d}^\top\boldsymbol{C}^{-1}\boldsymbol{d} \\
=& \underbrace{\log\frac{|\boldsymbol{C}|}{|\boldsymbol{C}_{\text{gen}}|} - b + \mathrm{tr}\left(\left(\boldsymbol{C}^{-1}\circ\boldsymbol{C}_{\text{gen}}\right)\mathbf{1}\right)}_{2\mathrm{KL}(p_c\|q_c)} \underbrace{- \sum_{i=1}^{b}\log\boldsymbol{r}_i^2 + \boldsymbol{d}^\top\boldsymbol{d} + \boldsymbol{r}^\top\boldsymbol{r} - b}_{-2\mathrm{KL}(p_m\|q_m)} \\
&+ \underbrace{\mathrm{tr}\left(\left(\boldsymbol{C}^{-1}\circ\boldsymbol{C}_{\text{gen}}\right)\left(\boldsymbol{r}\boldsymbol{r}^\top - \mathbf{1}\right)\right)}_{①} + \underbrace{\mathrm{tr}\left(\left(\boldsymbol{C}^{-1} - \mathbb{I}\right)\boldsymbol{d}\boldsymbol{d}^\top\right)}_{②} + \underbrace{\left(b - \boldsymbol{r}^\top\boldsymbol{r}\right)}_{③}.
\end{aligned} \tag{10}$$

Therefore, we have

$$2\left|\mathrm{KL}\left(p_c\|q_c\right) - \mathrm{KL}\left(p\|q\right)\right| \le \underbrace{2\mathrm{KL}\left(p_m\|q_m\right)}_{2\xi} + \left|①\right| + \left|②\right| + \left|③\right|, \tag{11}$$

Given that the marginal KL divergence is upper bounded by,

$$2\mathrm{KL}\left(p_m\|q_m\right) = -\sum_{i=1}^{b}\log\boldsymbol{r}_i^2 + \boldsymbol{d}^\top\boldsymbol{d} + \boldsymbol{r}^\top\boldsymbol{r} - b \le 2\xi, \tag{12}$$

and since $\forall x, x - 1 - \log x \ge 0$, we have

$$0 \le -\sum_{i=1}^{b}\log\boldsymbol{r}_i^2 + \boldsymbol{r}^\top\boldsymbol{r} - b \le 2\xi. \tag{13}$$

Then $\forall i$, $\boldsymbol{r}_i^2 - \log\boldsymbol{r}_i^2 - 1 \le 2\xi$, which means $\boldsymbol{r}_i = 1 + \mathcal{O}(\sqrt{\xi})$. As a result, we have the following bounds,

$$\|\boldsymbol{d}\boldsymbol{d}^\top\|_F = \boldsymbol{d}^\top\boldsymbol{d} \le 2\xi, \tag{14}$$

$$\left|b - \boldsymbol{r}^\top\boldsymbol{r}\right| = \mathcal{O}(b\sqrt{\xi}), \tag{15}$$

$$\|\boldsymbol{r}\boldsymbol{r}^\top - \mathbf{1}\|_F = \mathcal{O}(b\sqrt{\xi}). \tag{16}$$

We further let $\lambda := \lambda_{min}(\boldsymbol{C})$ be the smallest eigenvalue of $\boldsymbol{C}$. Then, we have $\|\boldsymbol{C}^{-1}\|_2 = \frac{1}{\lambda}$. Because $\boldsymbol{C}_{\text{gen}}$ is a correlation matrix, $\|\boldsymbol{C}_{\text{gen}}\|_\infty = 1$. Because

$$\left(\mathrm{tr}\left(\boldsymbol{A}^\top\boldsymbol{B}\right)\right)^2 \le \mathrm{tr}\left(\boldsymbol{A}^\top\boldsymbol{A}\right)\mathrm{tr}\left(\boldsymbol{B}^\top\boldsymbol{B}\right) = \|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2, \tag{17}$$

which gives us the upper bound of $\left|①\right|$:

$$\begin{aligned}
\left|①\right| =& \left|\mathrm{tr}\left(\left(\boldsymbol{C}^{-1}\circ\boldsymbol{C}_{\text{gen}}\right)\left(\boldsymbol{r}\boldsymbol{r}^\top - \mathbf{1}\right)\right)\right| \\
\le& \|\boldsymbol{C}^{-1}\circ\boldsymbol{C}_{\text{gen}}\|_F\|\boldsymbol{r}\boldsymbol{r}^\top - \mathbf{1}\|_F \\
\le& \|\boldsymbol{C}^{-1}\|_F\|\boldsymbol{r}\boldsymbol{r}^\top - \mathbf{1}\|_F \\
\le& \frac{\sqrt{b}}{\lambda}\|\boldsymbol{r}\boldsymbol{r}^\top - \mathbf{1}\|_F, \\
=& \frac{\sqrt{b}}{\lambda}\mathcal{O}(b\sqrt{\xi}).
\end{aligned} \tag{18}$$

Similarly, we can further bound $\left|\textcircled{2}\right|$ by:

$$\begin{aligned}\left|\textcircled{2}\right| =& \left|\mathrm{tr}\left(\left(\boldsymbol{C}^{-1}-\mathbb{I}\right)\boldsymbol{d}\boldsymbol{d}^{\top}\right)\right| \\ \leq& \|\boldsymbol{C}^{-1}-\mathbb{I}\|_F\|\boldsymbol{d}\boldsymbol{d}^{\top}\|_F \\ \leq& \sqrt{2b+2\|\boldsymbol{C}^{-1}\|_F^2}\|\boldsymbol{d}\boldsymbol{d}^{\top}\|_F \\ \leq& \sqrt{2b+\frac{2b}{\lambda^2}}\|\boldsymbol{d}\boldsymbol{d}^{\top}\|_F \\ \leq& \sqrt{2b+\frac{2b}{\lambda^2}}2\xi. \end{aligned} \tag{19}$$

Lastly, we can bound:

$$\left|\textcircled{3}\right| = \left|b-\boldsymbol{r}^{\top}\boldsymbol{r}\right| = \mathcal{O}(b\sqrt{\xi}). \tag{20}$$

Overall, since $\xi \ll 1$, we have

$$\begin{aligned}2\left|\mathrm{KL}\left(p_c\|q_c\right)-\mathrm{KL}\left(p\|q\right)\right| \leq& 2\mathrm{KL}\left(p_m\|q_m\right)+\frac{\sqrt{b}}{\lambda}\mathcal{O}(b\sqrt{\xi})+\sqrt{2+\frac{2b}{\lambda^2}}2\xi+\mathcal{O}(b\sqrt{\xi}) \\ \leq& 2\xi+\frac{\sqrt{b}}{\lambda}\mathcal{O}(b\sqrt{\xi})+\sqrt{2+\frac{2b}{\lambda^2}}2\xi+\mathcal{O}(b\sqrt{\xi}) \\ =& \mathcal{O}\left(\frac{b^{3/2}}{\lambda}\sqrt{\xi}\right). \end{aligned} \tag{21}$$

□