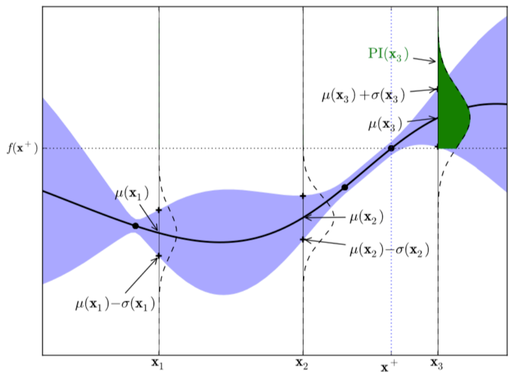
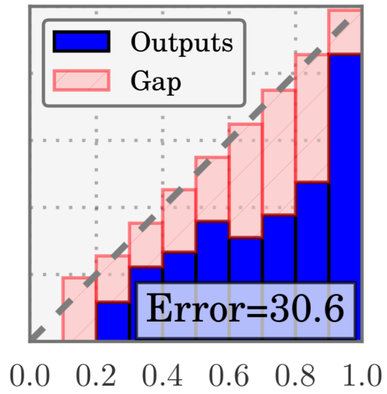
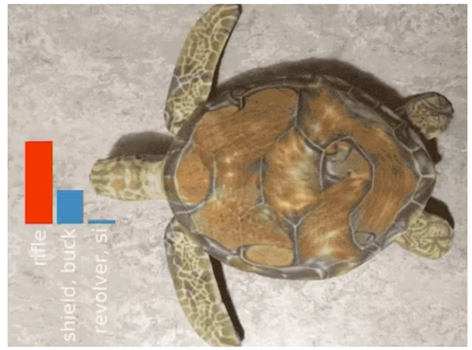
Beyond **Marginal Uncertainty:**



Exploration



Calibration

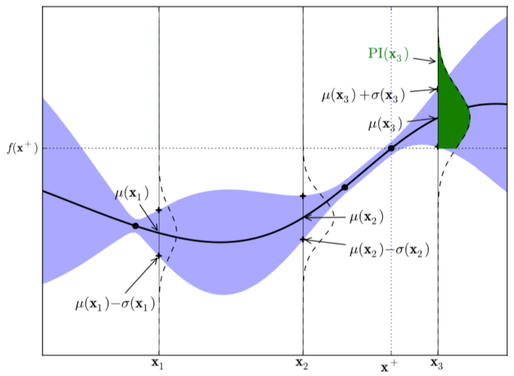


Adversarial Robustness

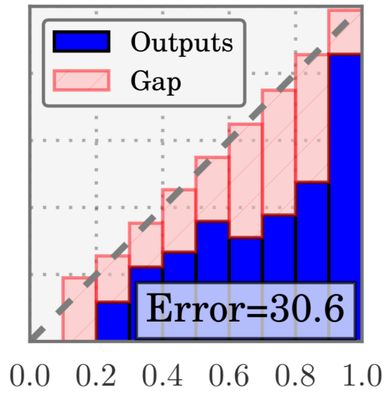
Chaoqi Wang*, **Shengyang Sun***, Roger Grosse

(Brochu et al., 2010) (Guo et al., 2017) (Athalye et al., 2017)

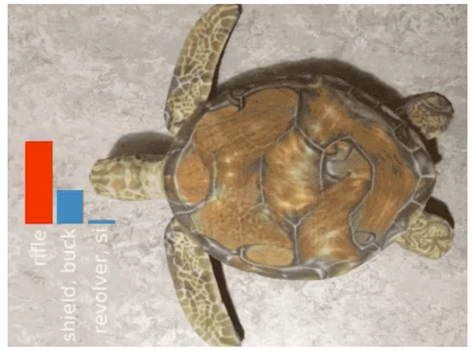
Beyond **Marginal Uncertainty:**



Exploration

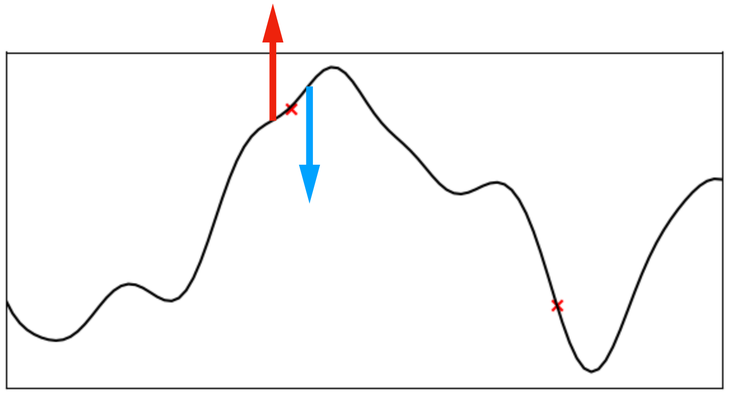


Calibration

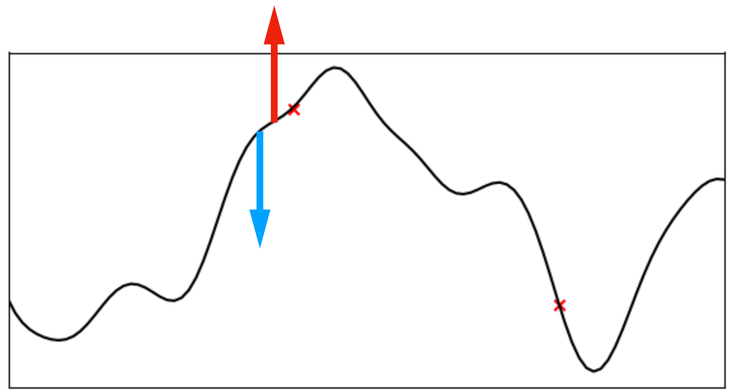


Adversarial Robustness

How Accurately can Bayesian Regression Models Estimate **Posterior Predictive Correlations?**



Negative Correlations



Positive Correlations

Chaoqi Wang*, **Shengyang Sun***, Roger Grosse

(Brochu et al., 2010) (Guo et al., 2017) (Athalye et al., 2017)

Posterior Predictive Correlations (PPCs)

- Posterior Predictive Means.

$$\mu_x = \mathbb{E}[f(x)|\mathcal{D}_{tr}]$$

- Posterior Predictive Variances.

$$\sigma_x^2 = \mathrm{Var}[f(x)|\mathcal{D}_{tr}]$$

- Posterior Predictive Covariances.

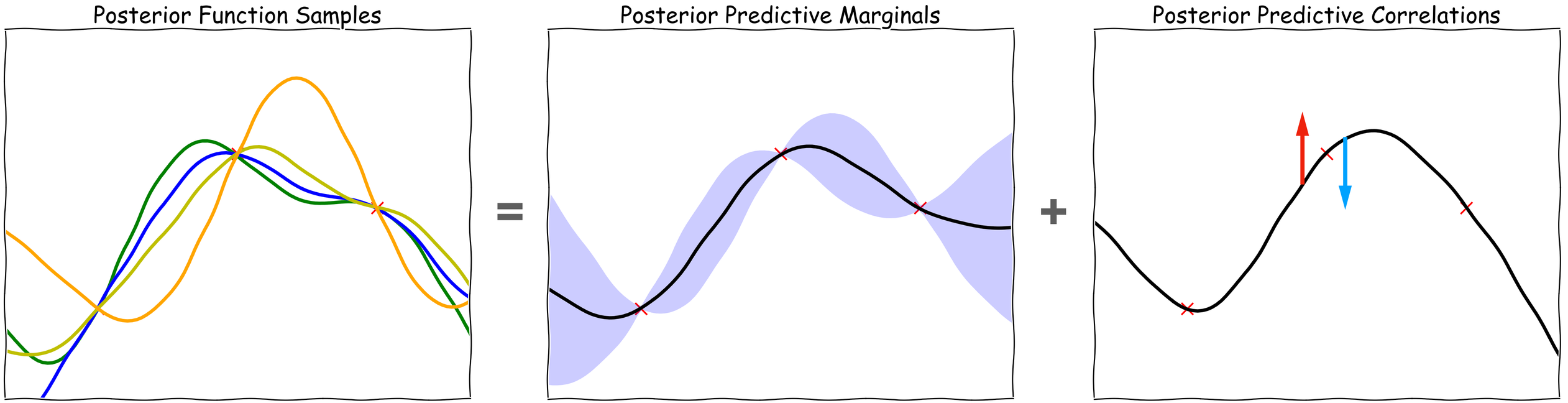
$$\Sigma(x, x') = \mathbb{E}[(f(x) - \mu_x)(f(x') - \mu_{x'})|\mathcal{D}_{tr}]$$

- Posterior Predictive Correlations.

$$\rho(f(x), f(x')|\mathcal{D}_{tr}) = \Sigma(x, x')/(\sigma_x \sigma_{x'}),$$

Why caring about PPCs?

Posteriors = Marginals + Correlations



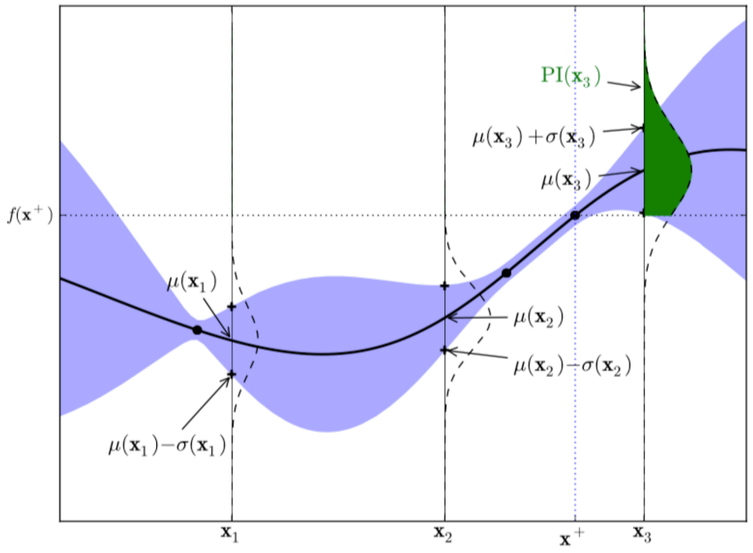
Many algorithms rely on the full posterior instead of merely on marginals:

(/ Predictive / Max-value) Entropy Search; Thompson Sampling; Knowledge Gradient; …

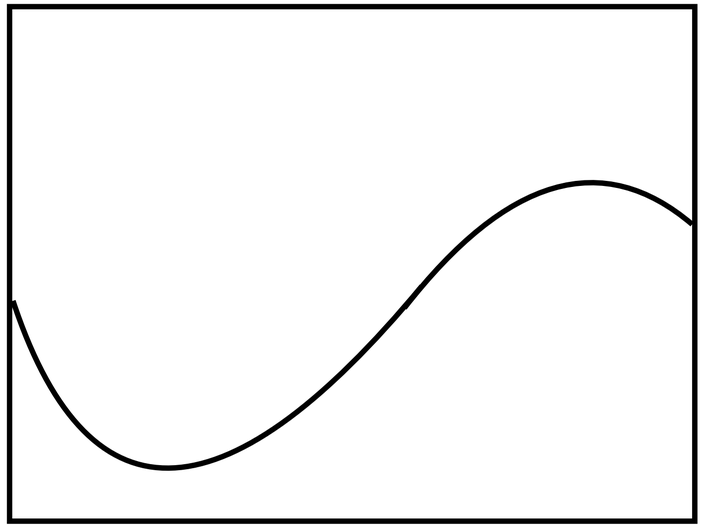
PPCs enable efficient Explorations

- Bayesian Optimization (BO) aims to quickly navigate to the global maximum.

Function Value



Evaluation Cost

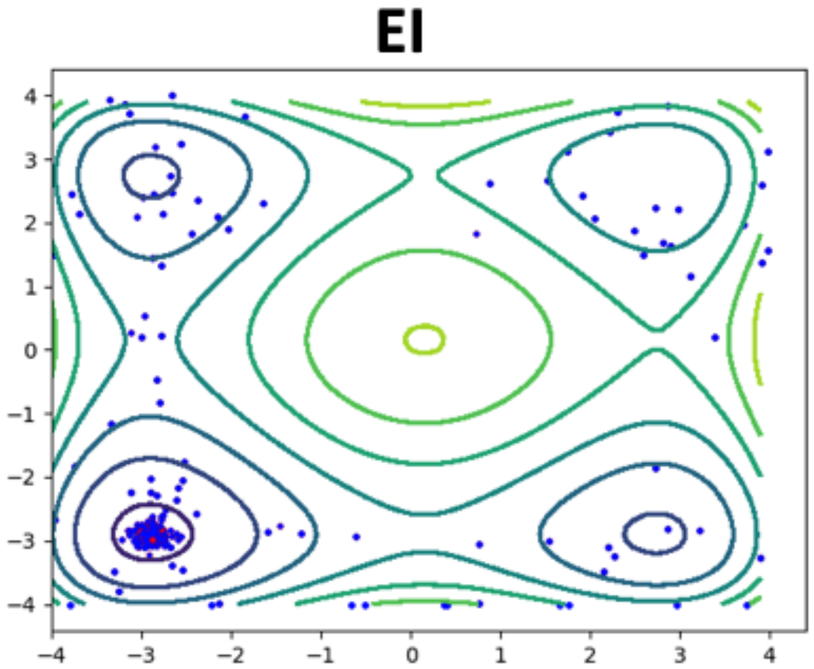


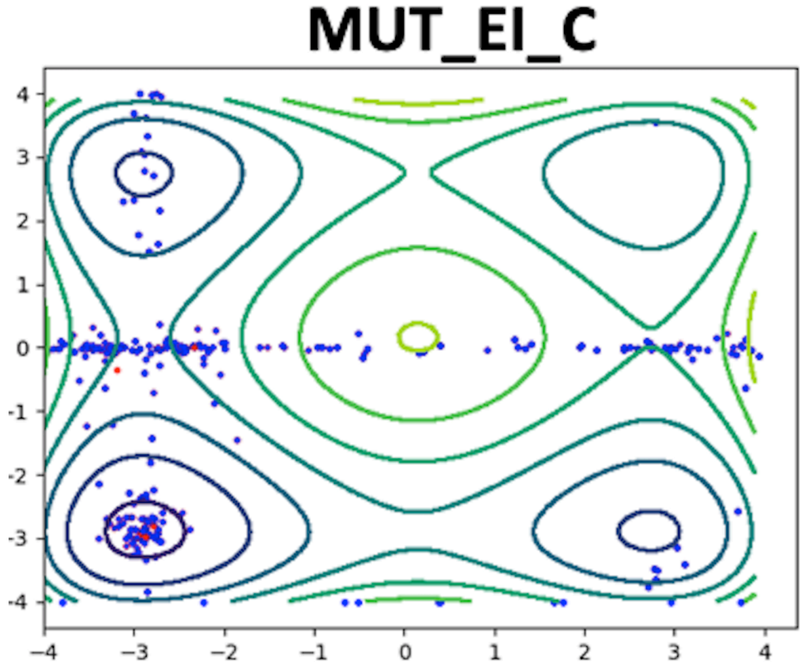
- What if each function evaluation comes with a cost ?
  - Elapsed Time for one Hyper-parameter Training
  - Money costed for one Protein Lab Experiment

(Brochu et al., 2010)

PPCs enable efficient Explorations

- Evaluating "low-cost regions" can provide information for "high-cost regions"

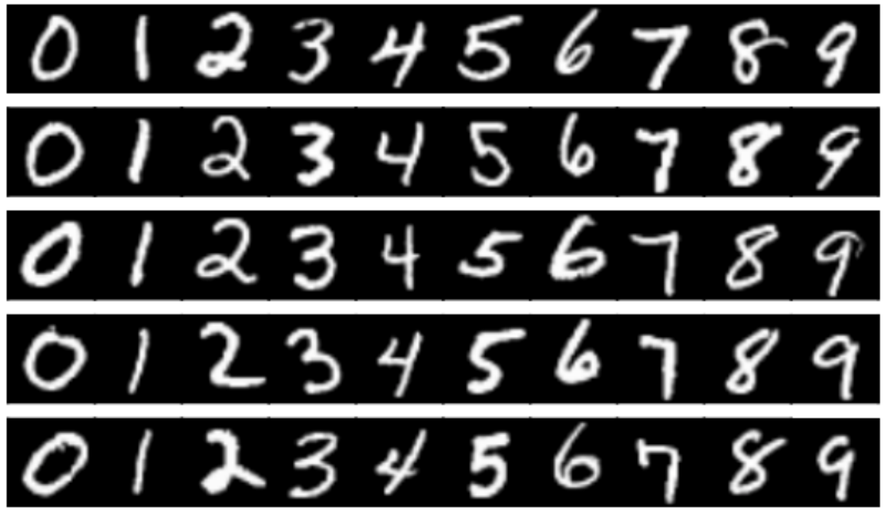




PPCs enable Data Summarizations

- Data summarization searches for a small set representative of a large dataset
  - Lower storage burden, Lower computational costs
  - Coresets, Inducing points, Replaying pseudo-data

Random Inducing Points



Optimized Inducing Points

(Titsias et al., 2019)

How to Evaluate PPC Estimations?

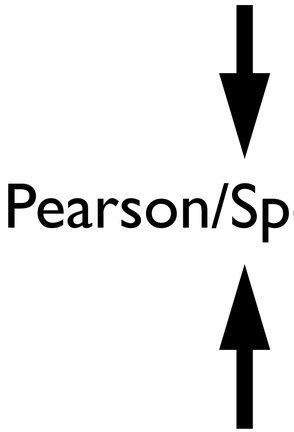
MetaCorrelations

Metacorrelations

If we have an Oracle Model

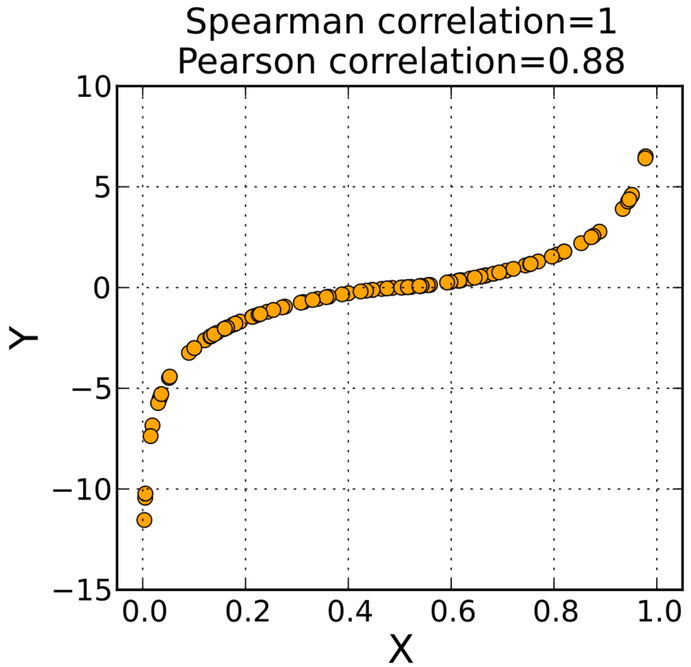
- Comparing with ground-truths is the standard proxy for evaluating models: accuracy, BELU score, mean squared error, …

Ground-Truth Correlations



Pearson/Spearman

Predictive Correlations



- Only available for synthetic data, but it is a useful tool for validating other metrics which can be computed on real data.

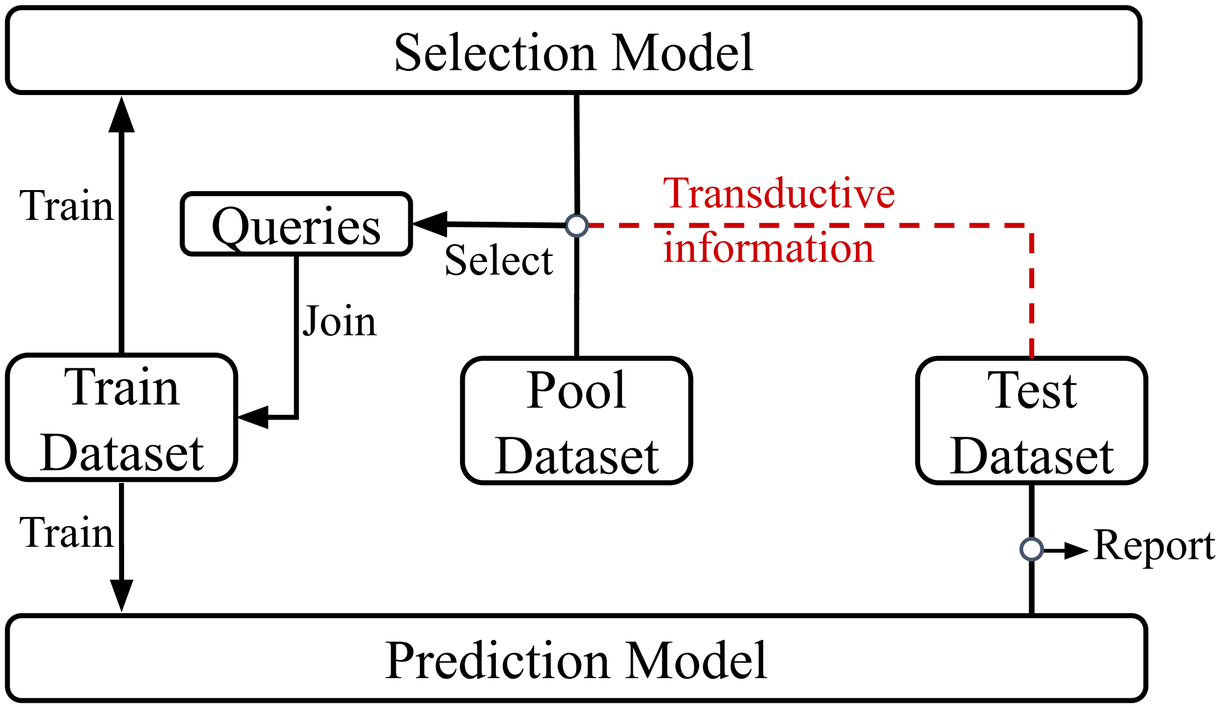
How to Evaluate PPC Estimations?

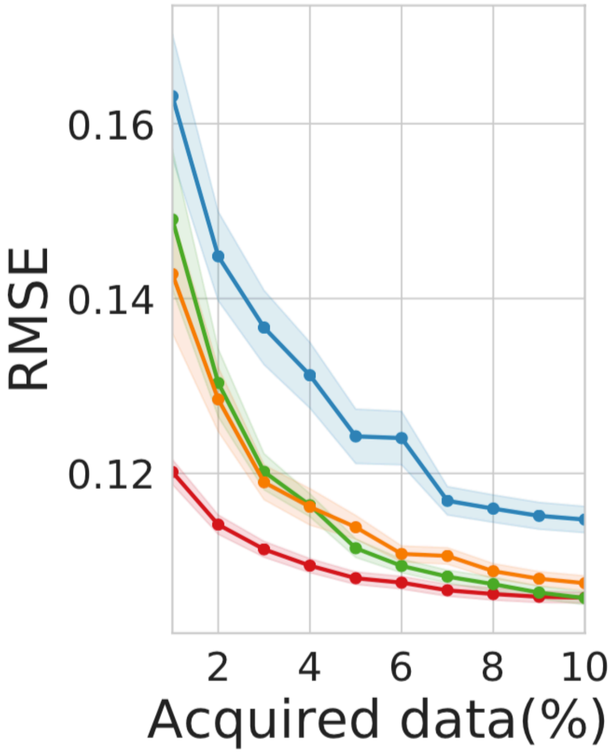
Transductive Active Learning

Transductive Active Learning (TAL)

A downstream task

- Active Learning (AL) allows the algorithm to choose training data interactively.





- TAL assumes the knowledge of the test set.

Transductive Active Learning (TAL)

A downstream task

The Acquisition Function

Total Information Gain (TIG)

$$\mathrm{TIG}(\mathbf{x}) = \mathbb{MI}[\mathbf{w}; y_{\mathbf{X}}]$$

Depends only on marginal uncertainty

Inclined to select points on the boundary

Transductive

Marginal Information Gain (MIG)

$$\mathrm{MIG}(\mathbf{x}) = \mathbb{MI}[f(\mathbf{x}_u); y_{\mathbf{X}}]$$

Transductive information for the test set

Naively choosing high-MIG points neglects the "diversity"

Query points in batch

Batched Marginal Information Gain (BatchMIG)

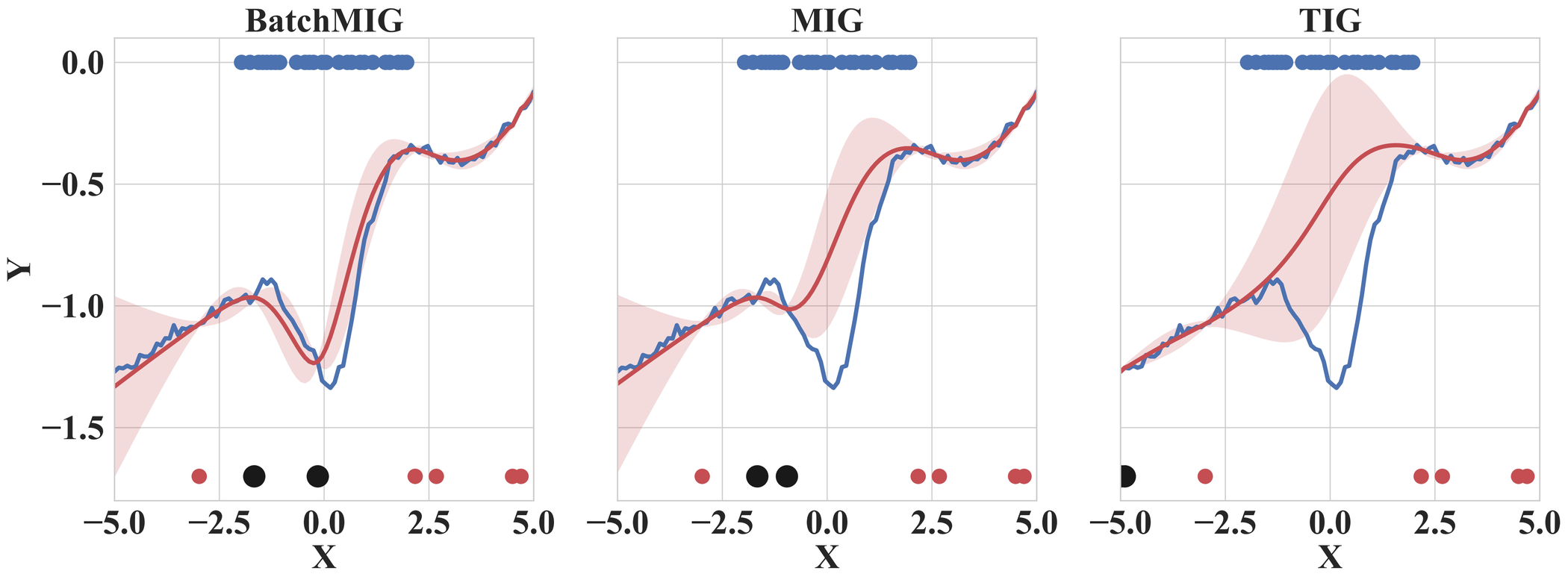
$$\mathrm{BatchMIG}(\mathbf{x}_{1:q}) = \mathbb{MI}[f(\mathbf{x}_u); y_{1:q}]$$

Transductive information for the test set

"Diversity" in a batch

Transductive Active Learning (TAL)

A downstream task

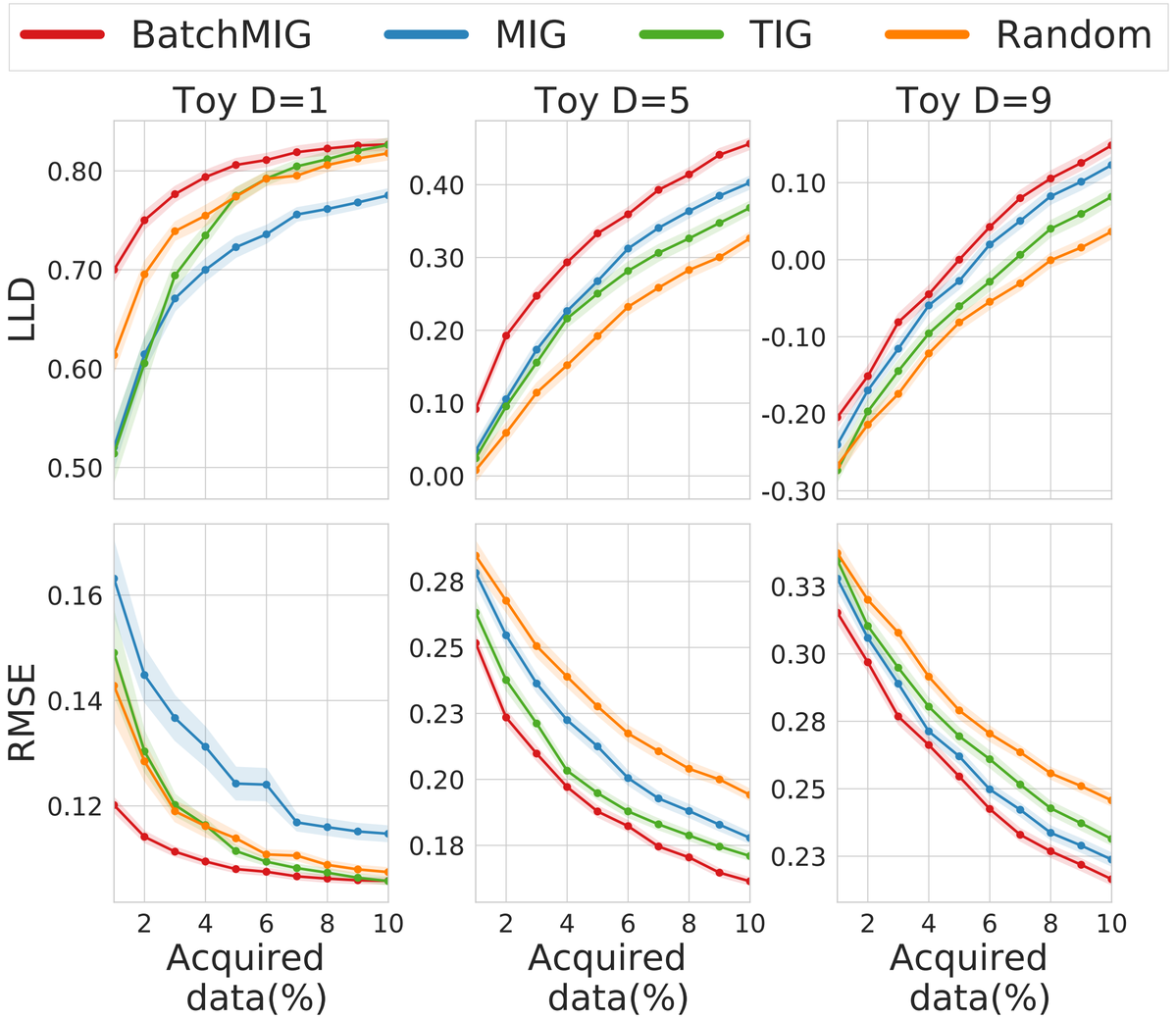


red curve: the prediction after query; blue curve: the ground-truth;
the testing points, the training points; the queried points;

Transductive Active Learning (TAL)

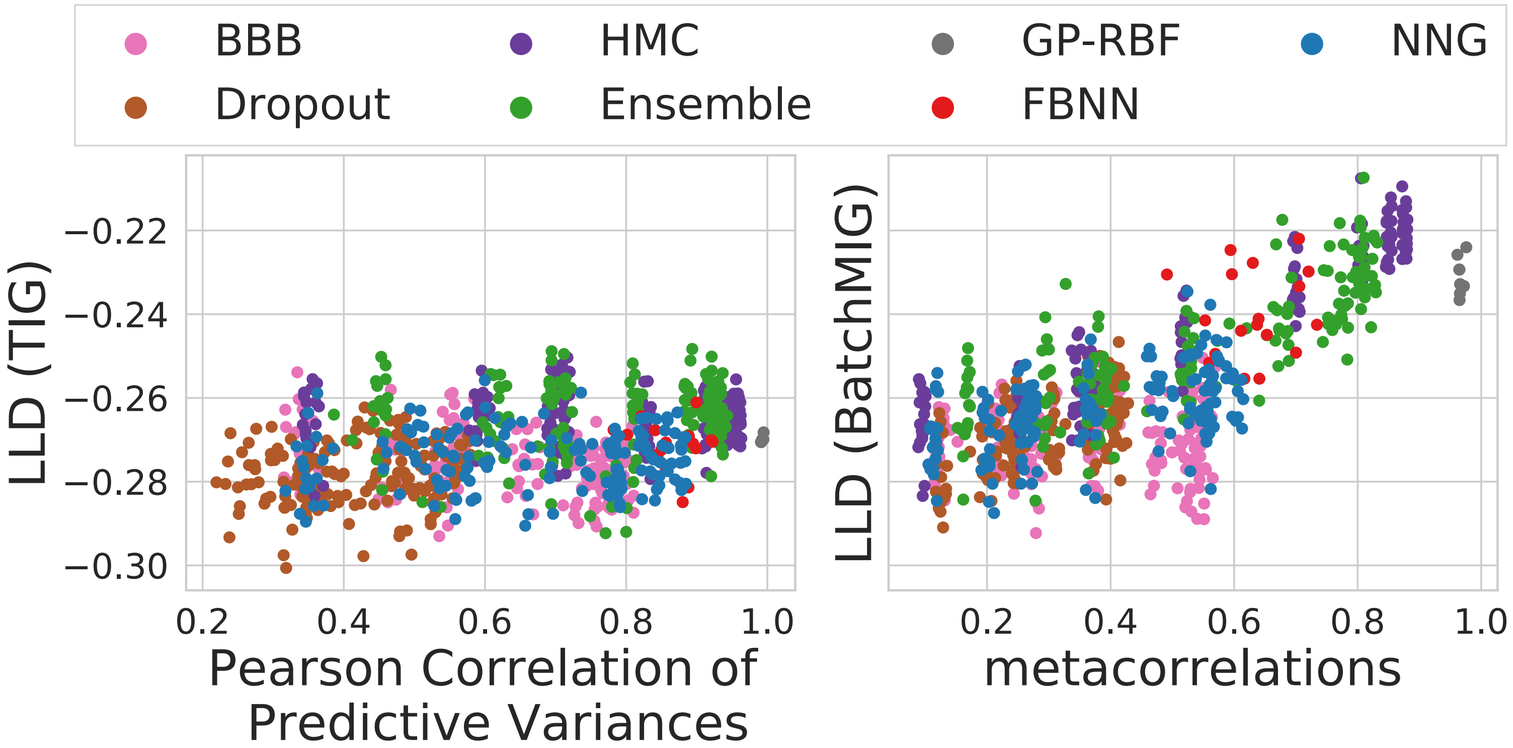
A downstream task

- BatchMIG improves data efficiency.



Transductive Active Learning (TAL)

A downstream task



- BatchMIG improves data efficiency compared to TIG.
- TAL performance is positively correlated to the quality of PPCs.
- TAL requires training the model multiple times in succession, which can be computationally prohibitive.

How to Evaluate PPC Estimations?

Cross-Normalized Log Likelihoods

Cross-normalized Log Likelihoods (XLL)

An efficient metric

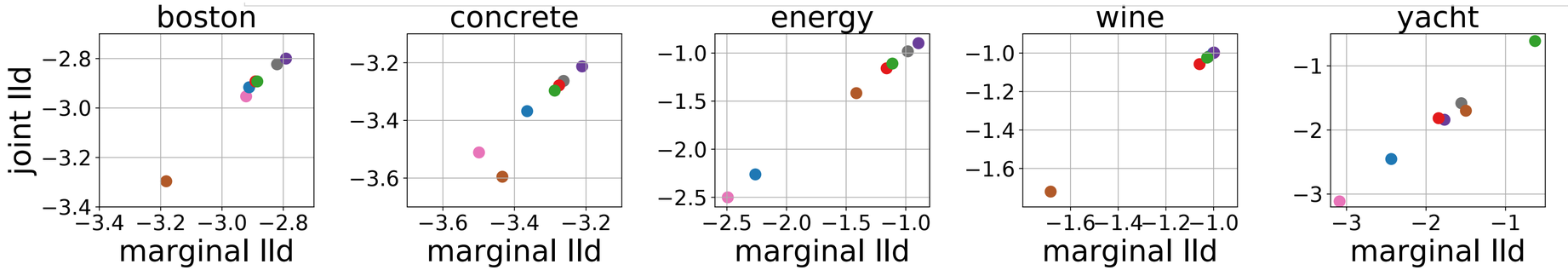
- Log marginal likelihood is a standard metric for evaluating marginals.

$$\sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i)$$

- How about using log joint likelihoods for evaluating PPCs ?

$$\log p(y_1, \ldots, y_b|\mathbf{x}_1, \ldots, \mathbf{x}_b)$$

  - Impact of Predictive Marginals



  - Uncorrelated Random Batches

Cross-normalized Log Likelihoods (XLL)

An efficient metric

- Impact of predictive marginals: Remove their impact by a *reference model*
- Cross-normalized Log Likelihoods (XLL)

$$\mathrm{XLL}(\mathbf{y}|\mathbf{X}, \mathcal{M}, \mathcal{M}_{\mathrm{ref}}) = \log \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_{\mathrm{ref}},\ \mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{ref}})\mathbf{C}_{\mathcal{M}}\mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{ref}}))$$

- XLL depends only on the predictive correlations $\mathbf{C}_{\mathcal{M}}$
- If the *reference marginals* approximate the ground truth well, then XLL reflects the quality of PPCs.

$$\mathrm{LogDet}(\boldsymbol{C}_{\mathrm{gt}}, \boldsymbol{C}_{\mathcal{M}}) = -\mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}\mathrm{XLL} + \mathcal{O}\left(\frac{b^{3/2}}{\lambda}\sqrt{\xi}\right) + c,$$

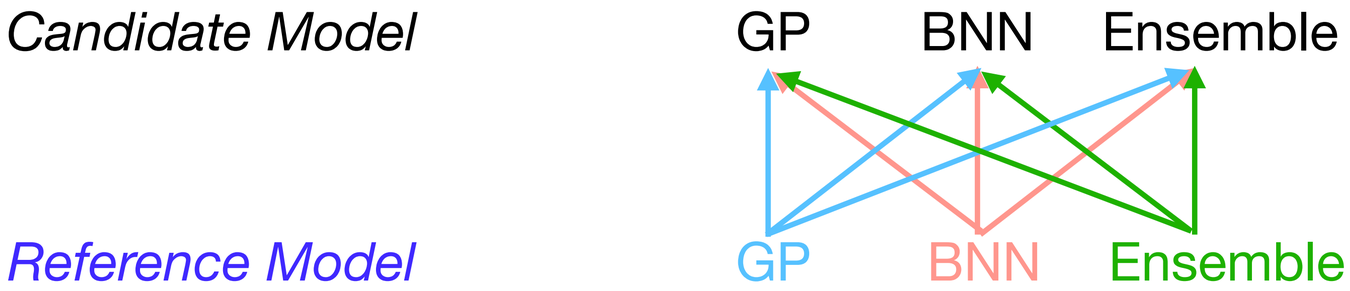
The quality of XLL

Approx Error of reference marginals

Cross-normalized Log Likelihoods (XLL)

An efficient metric

- Impact of predictive marginals: Remove their impact by a *reference model*
- Uncorrelated random batches: Select the top-correlated points using *reference model*
- The *reference model*: iterate the candidate models, and average the XLLs

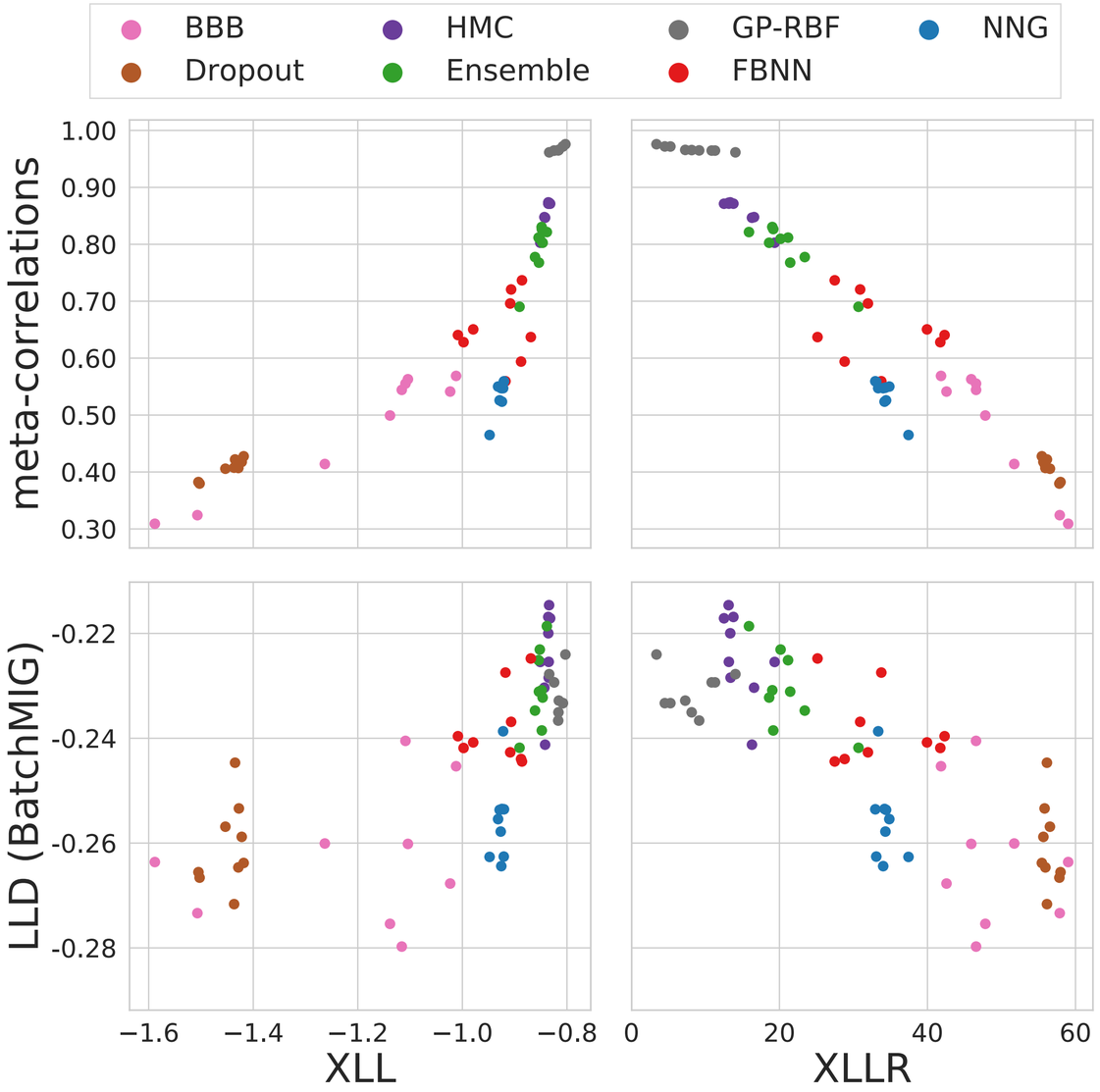


- XLLR: For each *reference model*, we compute the ranks for candidate models, and average the ranks under all *reference models*.

Cross-normalized Log Likelihoods (XLL)

An efficient metric

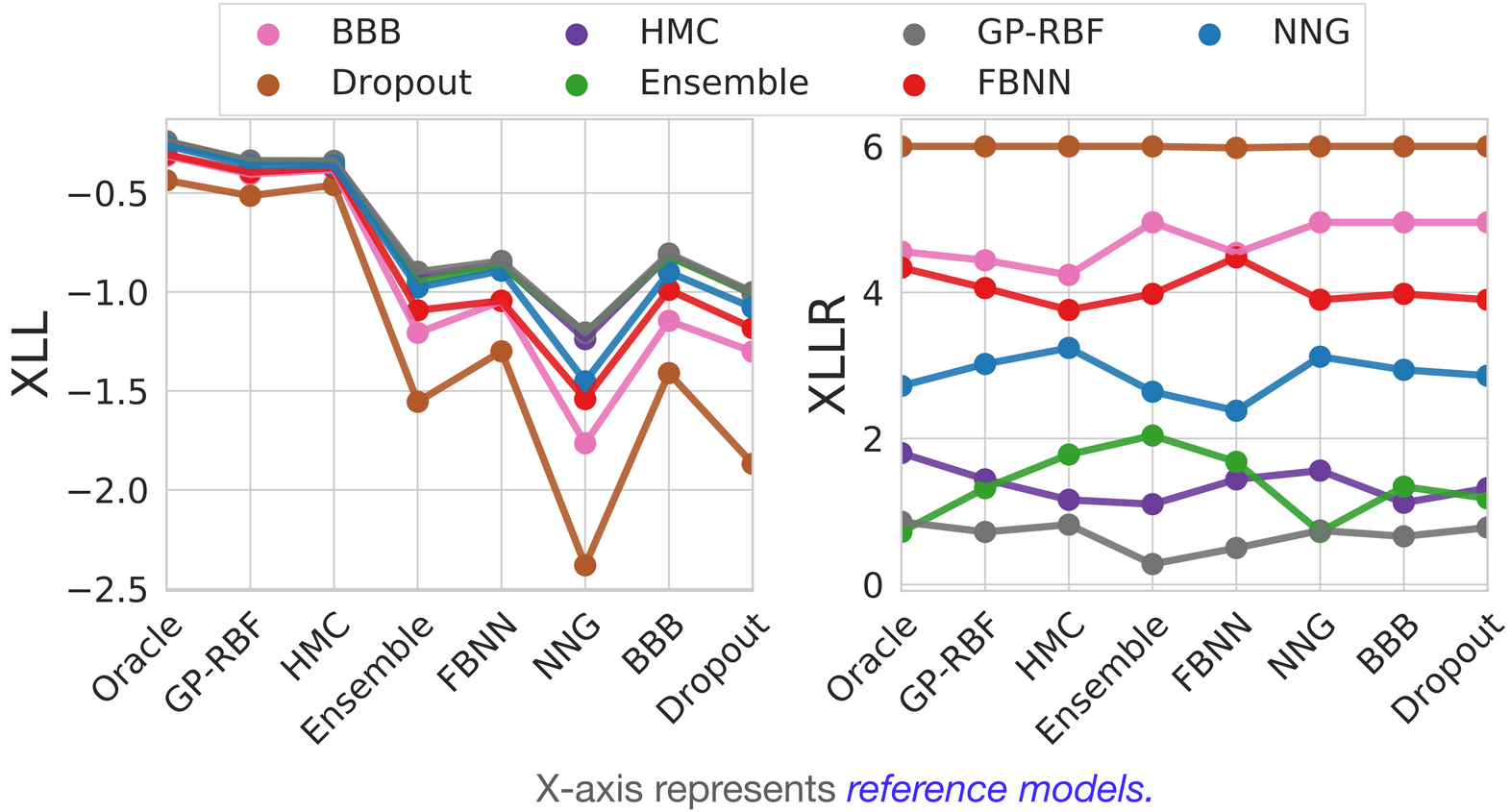
- XLL and XLLR are reliable proxy for metacorrelations and TAL performances.



Cross-normalized Log Likelihoods (XLL)

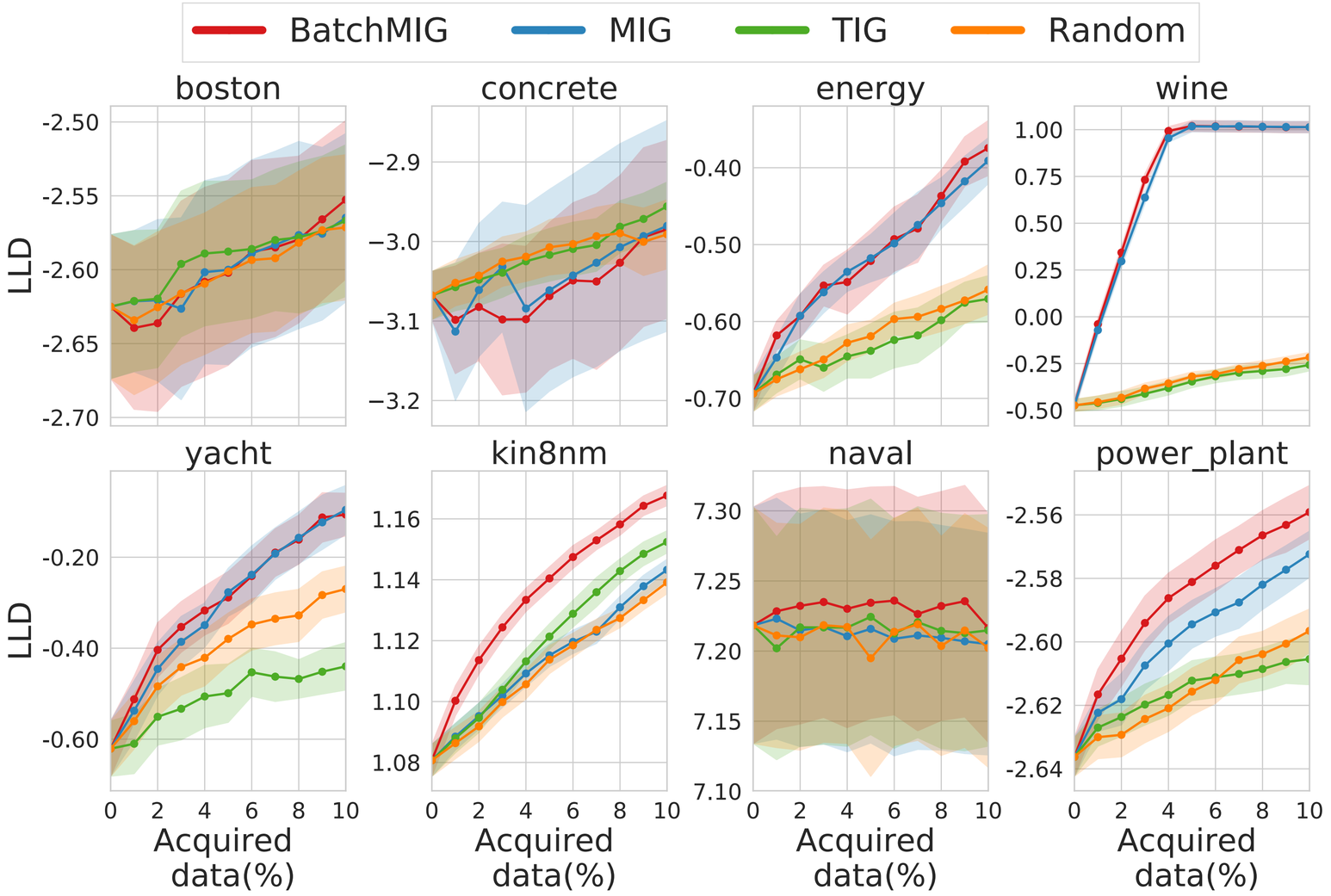
An efficient metric

- XLL and XLLR are robust of *reference models.*



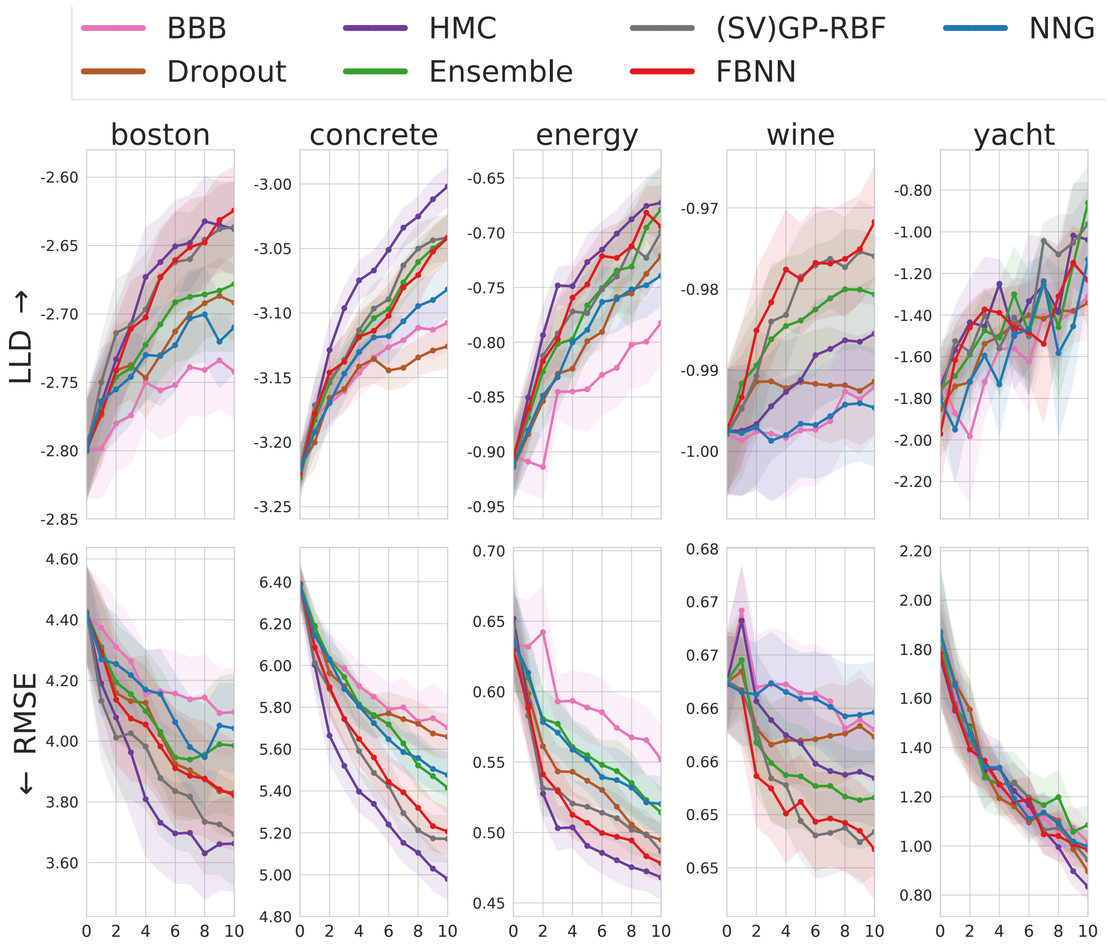
How Accurately can Bayesian Regression Models Estimate PPCs?

Transductive Active Learning

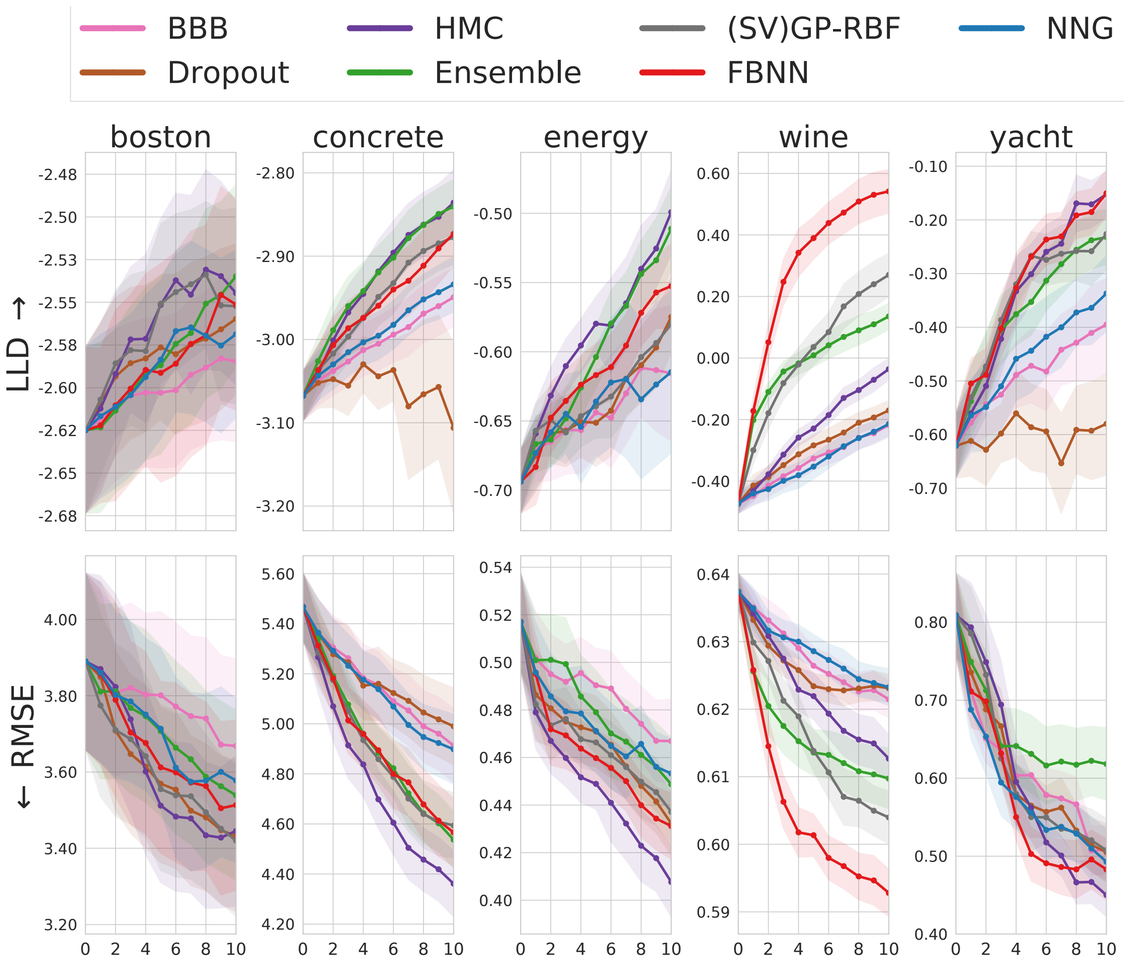


GP-NKN as the Prediction Model and the Selection Model

Transductive Active Learning

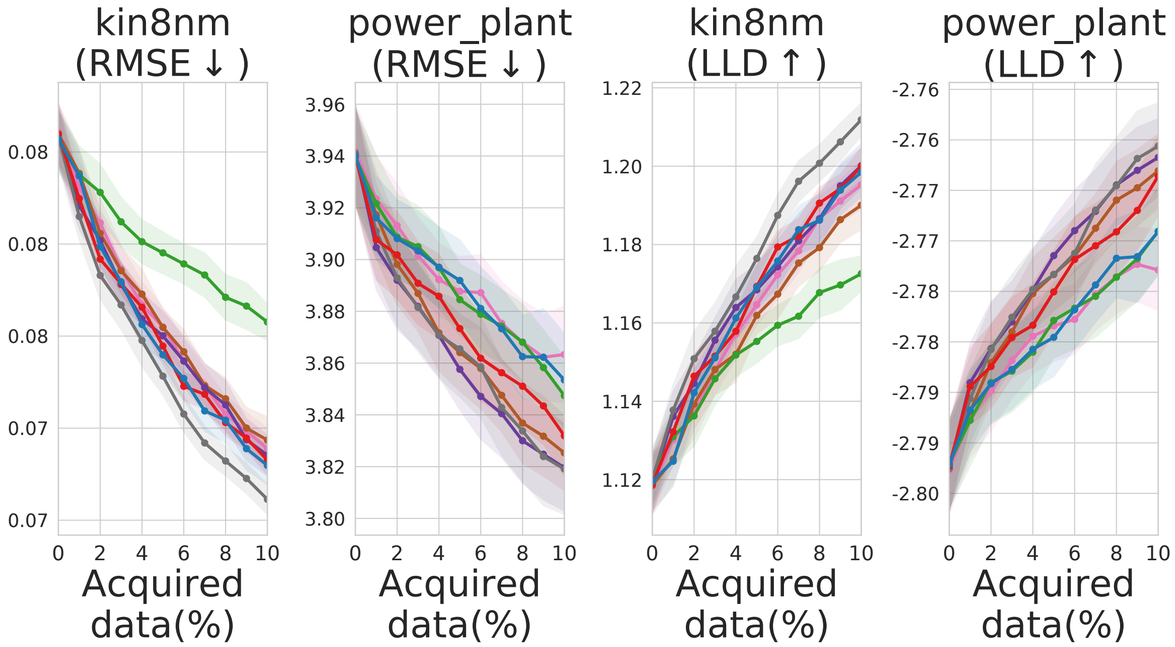


HMC as the Prediction Model

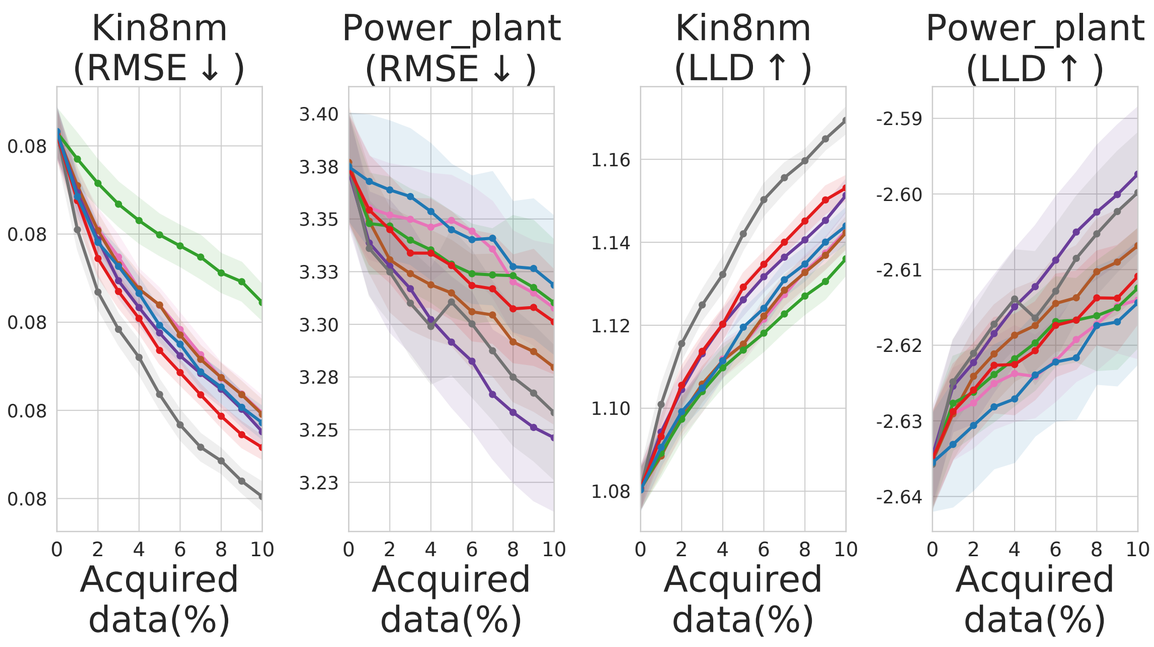


GP-NKN as the Prediction Model

Transductive Active Learning



HMC as the Prediction Model



GP-NKN as the Prediction Model

XLLR

Table 1: The average XLLR for each models on UCI datasets. We use color red to highlight the best ones (within one standard error), and color blue for the worst ones (within one standard error).

Dataset/Method	(SV)GP-RBF	BBB	NNG	HMC	FBNN	Dropout	Ensemble
Boston	2.53 (0.20)	4.24 (0.18)	2.57 (0.19)	0.93 (0.13)	3.20 (0.21)	5.31 (0.13)	2.21 (0.19)
Concrete	2.13 (0.14)	4.50 (0.14)	3.19 (0.17)	1.49 (0.18)	2.49 (0.14)	5.81 (0.09)	1.40 (0.21)
Energy	1.93 (0.17)	4.07 (0.19)	4.71 (0.15)	1.59 (0.22)	2.00 (0.17)	4.57 (0.15)	2.13 (0.21)
Wine	1.89 (0.18)	4.61 (0.13)	3.00 (0.16)	2.14 (0.15)	1.14 (0.21)	5.59 (0.12)	2.63 (0.18)
Yacht	1.90 (0.19)	3.74 (0.14)	3.49 (0.17)	2.63 (0.27)	1.76 (0.16)	5.71 (0.11)	1.77 (0.15)
Kin8nm	1.24 (0.11)	4.37 (0.07)	4.27 (0.16)	1.44 (0.15)	1.81 (0.15)	5.14 (0.25)	2.71 (0.18)
Naval	3.33 (0.14)	5.54 (0.12)	4.76 (0.08)	2.40 (0.18)	2.31 (0.07)	1.43 (0.28)	1.23 (0.12)
Power_plant	1.19 (0.11)	3.81 (0.12)	5.04 (0.15)	1.93 (0.15)	3.54 (0.11)	4.33 (0.29)	1.16 (0.15)
Mean ↓	2.02	4.36	3.88	1.82	2.28	4.73	1.91

Thanks for attending !

Takeaway: Three Metrics to Benchmark Posterior Predictive Correlations

Code repository: https://github.com/ssydasheng/predictive-correlation-benchmark

XLL Results

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)

TAL with various Prediction Models

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