Benchmarking Uncertainty Quantification for Protein Engineering

Kevin P. Greenman,[†] Ava P. Amini,[*,‡] and Kevin K. Yang[*,‡]

[†]*Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA*

[‡]*Microsoft Research New England, Cambridge, MA, USA*

E-mail: ava.amini@microsoft.com; yang.kevin@microsoft.com

Abstract

Machine learning sequence-function models for proteins could enable significant advances in protein engineering, especially when paired with state-of-the-art methods to select new sequences for property optimization and/or model improvement. Such methods (Bayesian optimization and active learning) require calibrated estimations of model uncertainty. While studies have benchmarked a variety of deep learning uncertainty quantification (UQ) methods on standard and molecular machine-learning datasets, it is not clear if these results extend to protein datasets. In this work, we implemented a panel of deep learning UQ methods on regression tasks from the Fitness Landscape Inference for Proteins (FLIP) benchmark. We compared results across different degrees of distributional shift using metrics that assess each UQ method's accuracy, calibration, coverage, width, and rank correlation. Additionally, we compared these metrics using one-hot encoding and pretrained language model representations, and we tested the UQ methods in a retrospective active learning setting. These benchmarks enable us to provide recommendations for more effective design of biological sequences using machine learning.

Abbreviations

ML, UQ, CNN, FLIP, GP, BO, BRR, MVE, SVI, AAV, GB1, RMSE, MAE, RQ

Keywords

uncertainty quantification, protein engineering, active learning

1 Introduction

Machine learning (ML) has already begun to accelerate the field of protein engineering by providing low-cost predictions of phenomena that require time- and resource-intensive labeling by experiments or physics-based simulations (*1*). It is often necessary to have an estimate of model uncertainty in addition to the property prediction, as the performance of an ML model can be highly dependent on the domain shift between its training and testing data (*2*). Because protein engineering data is often collected in a manner that violates the independent and identically distributed (i.i.d.) assumptions of many ML approaches, (*3*), tailored ML methods are required to guide the selection of new experiments from a protein landscape. Uncertainty quantification (UQ) can inform the selection of experiments in order to improve a ML model or optimize protein function through active learning or Bayesian optimization.

In chemistry and materials science, several studies have benchmarked common UQ methods against one another on standard datasets and have used or developed appropriate metrics to quantify the quality of these uncertainty estimates (*4*–*9*). These works have illustrated that the best choice of UQ method can depend on the dataset and other considerations such as representation and scaling. While some protein engineering work has leveraged uncertainty estimates, these studies have been mostly limited to single UQ methods such as convolutional neural network (CNN) ensembles (*10*) or Gaussian processes (GPs) (*11*, *12*).

Gruver et al. compared CNN ensembles to GPs (using traditional representations and pre-trained BERT (*13*) language model embeddings) in Bayesian optimization (BO) tasks (*14*). They found that CNN ensembles are often more robust to distribution shift than other types of models. Additionally, they report that most model types have more poorly calibrated uncertainties on out-of-domain samples. However, more comprehensive study of CNN UQ methods other than ensembles against GPs using a variety of uncertainty quality metrics has not yet been done. A comparison of uncertainty methods on different protein representations (e.g., one-hot encodings or embeddings from protein language models) in an active learning setting is also lacking.

In this work, we used a set of standardized, public protein datasets to evaluate a panel of UQ methods for protein sequence-function prediction (Figure 1). Our chosen datasets included splits with varied degrees of domain extrapolation, which enabled method evaluation in a setting similar to what might be experienced while collecting new experimental data for protein engineering. We assessed each model using a variety of metrics that captured different aspects of desired performance, including accuracy, calibration, coverage, width, and rank correlation. Additionally, we compared the performance of the UQ methods on one-hot encoded sequence representations and on embeddings computed from the ESM-1b protein masked language model (*15*). We find that the quality of UQ estimates are dependent on the landscape, task, and embedding, and that no single method consistently outperforms all others. Finally, we evaluated the UQ methods in an active learning setting with several acquisition functions, and demonstrated that uncertainty-based sampling often outperforms random sampling (especially in later stages of active learning), although better calibrated uncertainty does not necessarily equate to better active learning. The understanding gained from this work will enable more effective application of UQ techniques to machine learning in protein engineering.

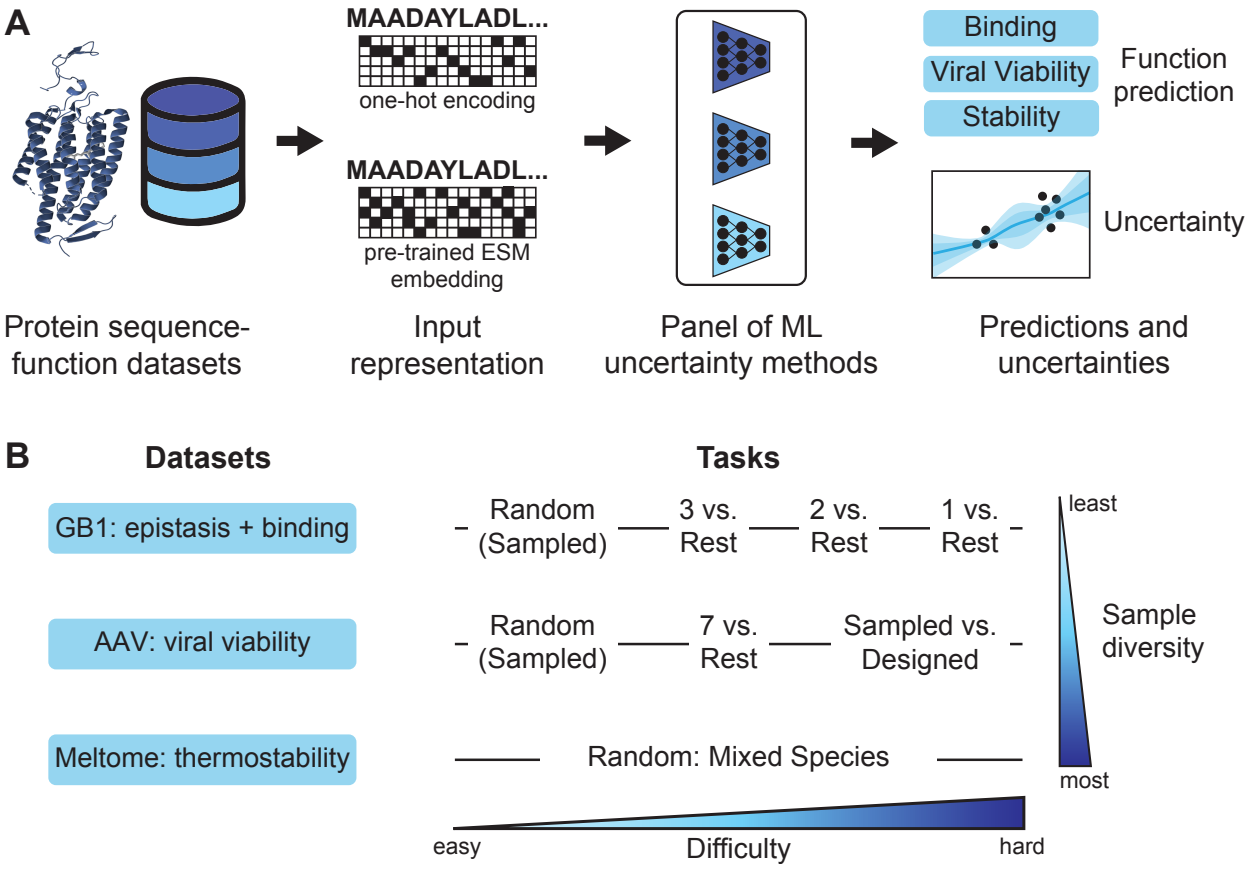

Figure 1: (A) Schematic of the approach for benchmarking uncertainty quantification (UQ) in machine learning for protein engineering. A panel of UQ methods were evaluated on protein fitness datasets to assess the quality of the uncertainty estimates and their utility in active learning. (B) Our study utilized three protein datasets/landscapes and different train-validation-test split tasks within each dataset. These datasets and tasks covered a range of sample diversities and domain shifts (task difficulties).

2 Results and Discussion

2.1 Uncertainty Quantification

Our first goal was to evaluate the calibration and quality of a variety of UQ methods. We implemented seven uncertainty methods for this benchmark: linear Bayesian ridge regression (BRR) (*16*, *17*), Gaussian processes (GPs) (*18*), and five methods using variations on a convolutional neural network (CNN) architecture. The CNN implementation from FLIP (*3*) provided the core architecture used by our dropout (*19*), ensemble (*20*), evidential (*21*), mean-variance estimation (MVE) (*22*), and last-layer stochastic variational inference (SVI)

(*23*) models. Additional model details are provided in Section 4.

The landscapes used in this work were taken from the Fitness Landscape Inference for Proteins (FLIP) benchmark (*3*). These include the binding domain of an immunoglobulin binding protein (GB1), adeno-associated virus stability (AAV), and thermostability (Meltome) data landscapes, which cover a large sequence space and a broad range of protein families. The FLIP benchmark includes several train-test splits, or tasks, for each landscape. Most of these tasks are designed to mimic common, real-world data collection scenarios and are thus a more realistic assessment of generalization than random train-test splits. However, random splits are also included as a point of reference. We chose 8 of the 15 FLIP tasks to benchmark the panel of uncertainty methods. We selected these tasks to be representative of several regimes of domain shift – random sampling with no domain shift (AAV/Random, Meltome/Random, and GB1/Random); the highest (and most relevant) domain-shift regimes (AAV/Random vs. Designed and GB1/1 vs. Rest); and less aggressive domain shifts (AAV/7 vs. Rest, GB1/2 vs. Rest and GB1/3 vs. Rest). The Datasets section of the Methods provides notes on the nomenclature used for these tasks.

We trained the seven models on each of the eight tasks described above and evaluated their performance on the test set using the metrics described in Section 4.7. We compare model calibration and accuracy in Figure 2 and the percent coverage versus average width relative to range in Figure 3. These figures illustrate the results for models trained on the embeddings from a pretrained ESM language model (*15*); the corresponding results using one-hot encodings are shown in Figures S1 and S2.

As expected, the splits with the least required domain extrapolation tend to have more accurate models (lower RMSE; Fig. 2). However, the relationship between miscalibration area and extrapolation is less clear; some models are highly calibrated on the most difficult (highest domain shift) splits, while others are poorly calibrated even on random splits. There is no single method that performs consistently well across splits and landscapes, but some trends can be observed. For example, ensembling is often one of the highest accuracy CNN

models, but also one of the most poorly calibrated. Additionally, GP and BRR models are often better calibrated than CNN models. For the AAV and GB1 landscapes (Fig. 2a, c), model miscalibration area usually increases slightly while RMSE increases more substantially with increasing domain shift.

In addition to accuracy and calibration, we assessed each method in terms of the coverage and width of its uncertainty estimates. A good uncertainty method results in high coverage (a large percentage of points where the true value falls within the 95% confidence region established by the uncertainty) while still maintaining a small average width. The latter is necessary because predicting a very large and uniform value of uncertainty for every point would result in good coverage, so coverage alone is not sufficient. Figure 3 illustrates that many methods perform relatively well in either coverage or width (corresponding to the the top and left limits of the plot, respectively), but few methods perform well in both. Similarly to Figure 2, there is some observable trend that more challenging splits are further from the optimal part (upper left) of the plot; this trend is more clear for the GB1 splits (Fig. 3b) than for the AAV splits. Most models trained on the AAV landscape (Fig. 3a) have a similar average width/range ratio for all splits, but for the GB1 landscape (Fig. 3c), this ratio typically increases as the domain shift increases. The locations of the sets of points for each model type shared some similarities across landscapes. CNN SVI often has low coverage and low width, CNN MVE often has moderate coverage and moderate width, and CNN Evidential and BRR often have high coverage and high width. The results for all prediction and uncertainty metrics are shown in Tables S1-S22.

We next assessed how target predictions and uncertainty estimates depended on the degree of domain shift. Across datasets and splits, we compared the ranking performance of each method in terms of predictions relative to true values and uncertainty estimates relative to true errors (ESM in Figure 4 and OHE in Figure S3). The splits are ordered according to domain shift within their respective landscapes (lowest to highest shift from left to right). We observe that the rank correlation of the predictions to the true labels

generally decreases moving from less to more domain shift within a landscape, consistent with expectation, with the exception of AAV/Random vs. Designed models performing better than AAV/7 vs. Rest models (Fig. 4a). Most methods exhibit similar performance in ρ within the same task. For many tasks, GP and BRR models perform as well or better than CNN models. Performance on ρ_{unc} is generally much worse than that on ρ, with some results showing negative correlation (Fig. 4b). MVE and evidential uncertainty methods are most performant in ρ_{unc} for most cases of low to moderate domain shift. Most methods have ρ_{unc} near zero for the most challenging splits. Despite the relatively good performance of MVE on tasks with low to moderate domain shift, it performs poorly in cases of high domain shift, which is consistent with its intended use as an estimator of aleatoric (data-dependent) uncertainty.

We find that the models trained on ESM embeddings outperform those trained on one-hot encodings in 21 out of 51 cases for rank correlation of test set predictions, and 29 out of 51 cases for rank correlation of test set uncertainties. The relative performance of the two representations on prediction and uncertainty rank correlation is shown in Figure S4. In terms of predictions, ESM embeddings often yield substantially better performance for tasks with high domain shift (e.g. GB1/1 vs. Rest and Meltome/Random), while OHE performs slightly better on tasks with lower domain shift (e.g. AAV/Random and GB1/3 vs. Rest). The relative uncertainty rank correlation performance, on the other hand, does not have a clear relationship to domain shift.

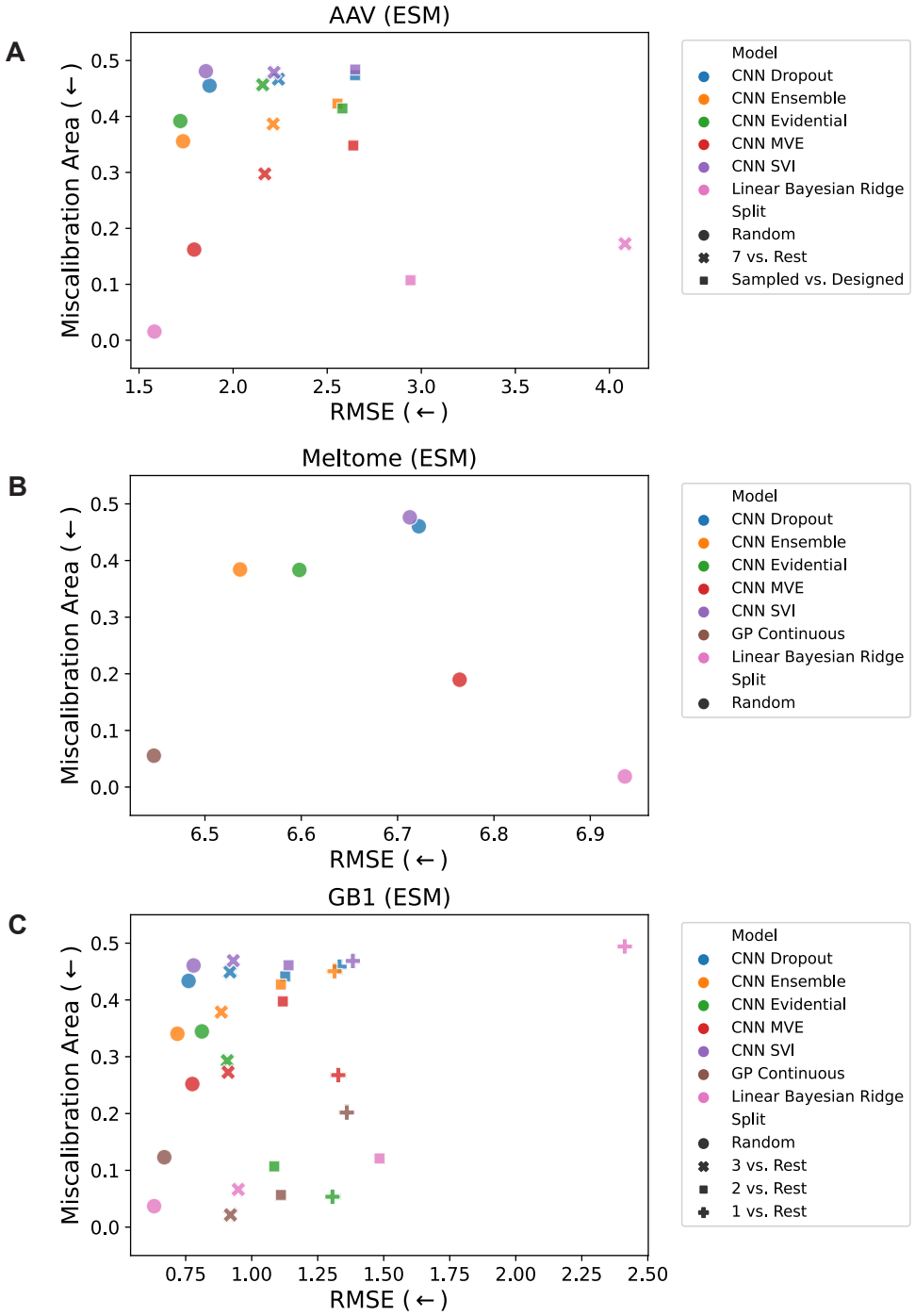

Figure 2: Miscalibration area vs. root mean square error (RMSE) for the (A) AAV, (B) Meltome, and (C) GB1 landscapes. Miscalibration area (also called the area under the calibration error curve or AUCE) quantifies the absolute difference between the calibration plot and perfect calibration. It is desirable to have a model that is both accurate and well-calibrated, so the best performing points are those closest to the lower left corner of the plots. Each point represents an average of 5 models trained using different random seeds for initialization of the CNN parameters and batching / stochastic gradient descent. The GP Continuous model is not shown for the AAV landscape due to memory constraints for training these models. Figure S1 shows the corresponding results for the OHE representation.

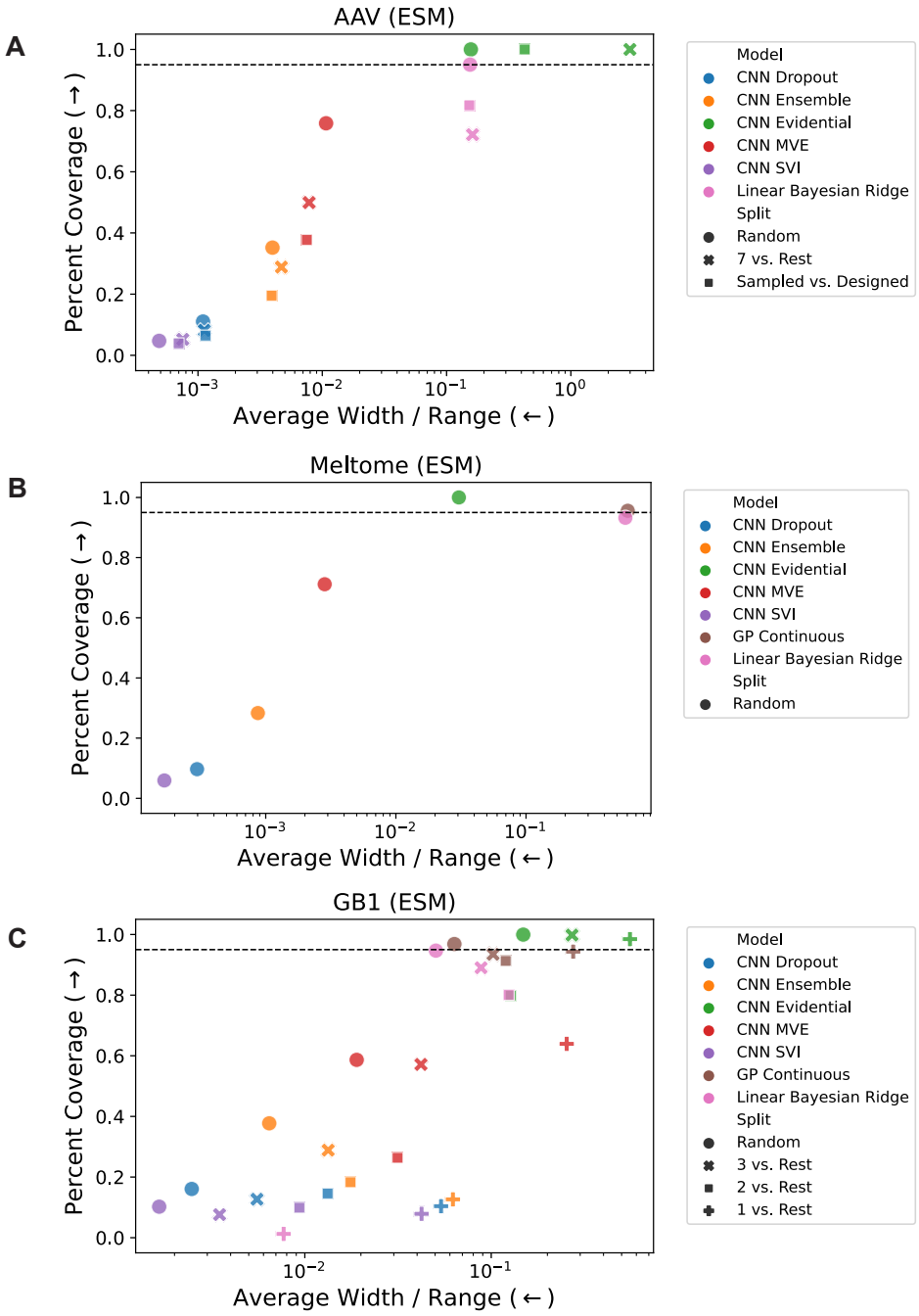

Figure 3: Coverage vs. average width / range for the (A) AAV, (B) Meltome, and (C) GB1 landscapes. Coverage is the percentage of true values that fall within the 95% confidence interval ($\pm 2\sigma$) of each prediction, and the width is the size of the 95% confidence region relative to the range of the training set ($4\sigma/R$ where R is the range of the training set). A good model exhibits high coverage and low width, which corresponds to the upper left of each plot. The horizontal dashed line indicates 95% coverage. Each point represents an average of 5 models trained using different random seeds for initialization of the CNN parameters and batching / stochastic gradient descent. The GP Continuous model is not shown for the AAV landscape due to memory constraints for training these models. Figure S2 shows the corresponding results for the OHE representation.

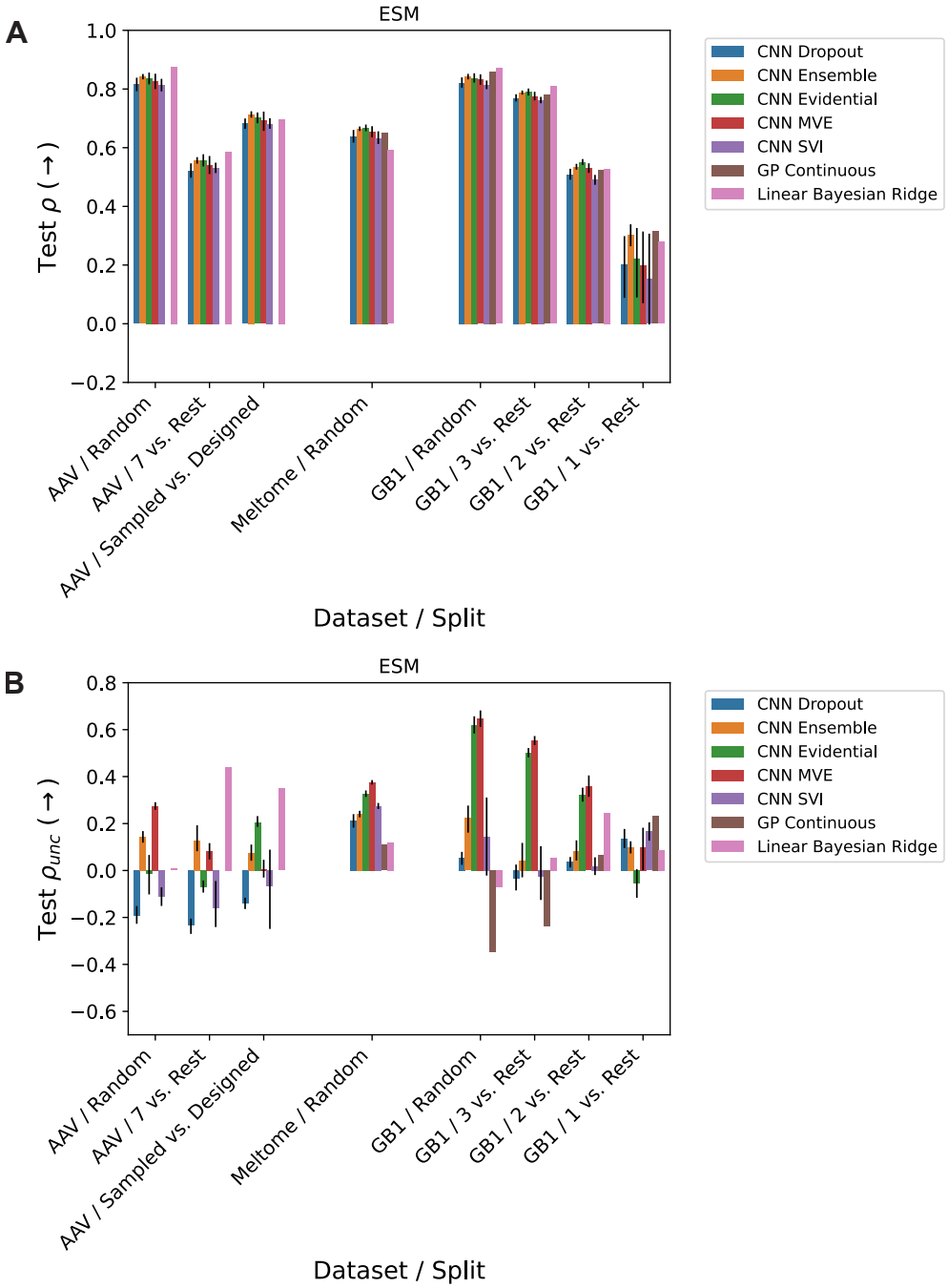

Figure 4: Spearman rank correlations of (A) predictions (ρ) and (B) uncertainties (ρ_{unc}) vs. extrapolation. Within each landscape (AAV, Meltome, and GB1), splits are qualitatively ordered by the amount of domain shift between train and test sets, with the lowest domain shift on the left and the highest domain shift on the right. Error bars on the CNN results represent the 95% confidence interval calculated from 5 different random seed for initialization of the CNN parameters and batching / stochastic gradient descent. Figure S3 shows the corresponding results for the OHE representation.

2.2 Active Learning

In protein engineering, the purpose of uncertainty estimation is typically to intelligently prioritize sample acquisition for experimentation. One such use case of uncertainty is in active learning, where uncertainty estimates are used to inform sampling with the goal of improving model predictions overall (i.e., to achieve an accurate model with less training data; Fig. 5a). Having assessed the calibration and accuracy of the panel of UQ methods above, we next evaluated whether uncertainty-based active learning could make the learning process more sample-efficient. Across all datasets and splits using the pretrained ESM embeddings, data acquisition was simulated as iterative selection from the data library according to a given sampling strategy (acquisition function; see Methods for details). The results are summarized in Figure 5 for Spearman rank correlation (ρ) on three methods and one split per landscape, and additional results are shown in the Figures S5-S57 for other metrics, uncertainty methods, and splits. Across most models, the performance difference between the start of active learning (10% of training data) and end of active learning (100% of training data) is relatively small, and many models begin to plateau in performance before reaching 100% of training data.

The "explorative greedy" and "explorative sample" acquisition functions (which sample based on uncertainty alone or sample randomly weighted by uncertainty, respectively) sometimes significantly outperform random sampling, but this is not true across all methods and landscapes (Fig. 5b-d). In some cases, the performance of the uncertainty-based sampling strategies also varies depending on the fraction of the total training data available to the model. For example, for the Meltome/Random split and CNN evidential model (Fig. 5c), explorative greedy sampling results in a *decrease* in model performance after the first round of active learning while the explorative sample strategy increases performance. By the fourth round of active learning for this task, the two explorative strategies significantly outperform random sampling. This indicates that in the early stages of active learning when a model's uncertainty estimates are poorly calibrated, it may be advantageous to sample with at least

some randomness included in an uncertainty-based acquisition function. Overall, the results indicate that uncertainty-informed active learning can outperform random sampling and thus lead to more accurate machine learning models with fewer training points needing to be measured (Fig. 5b-d).

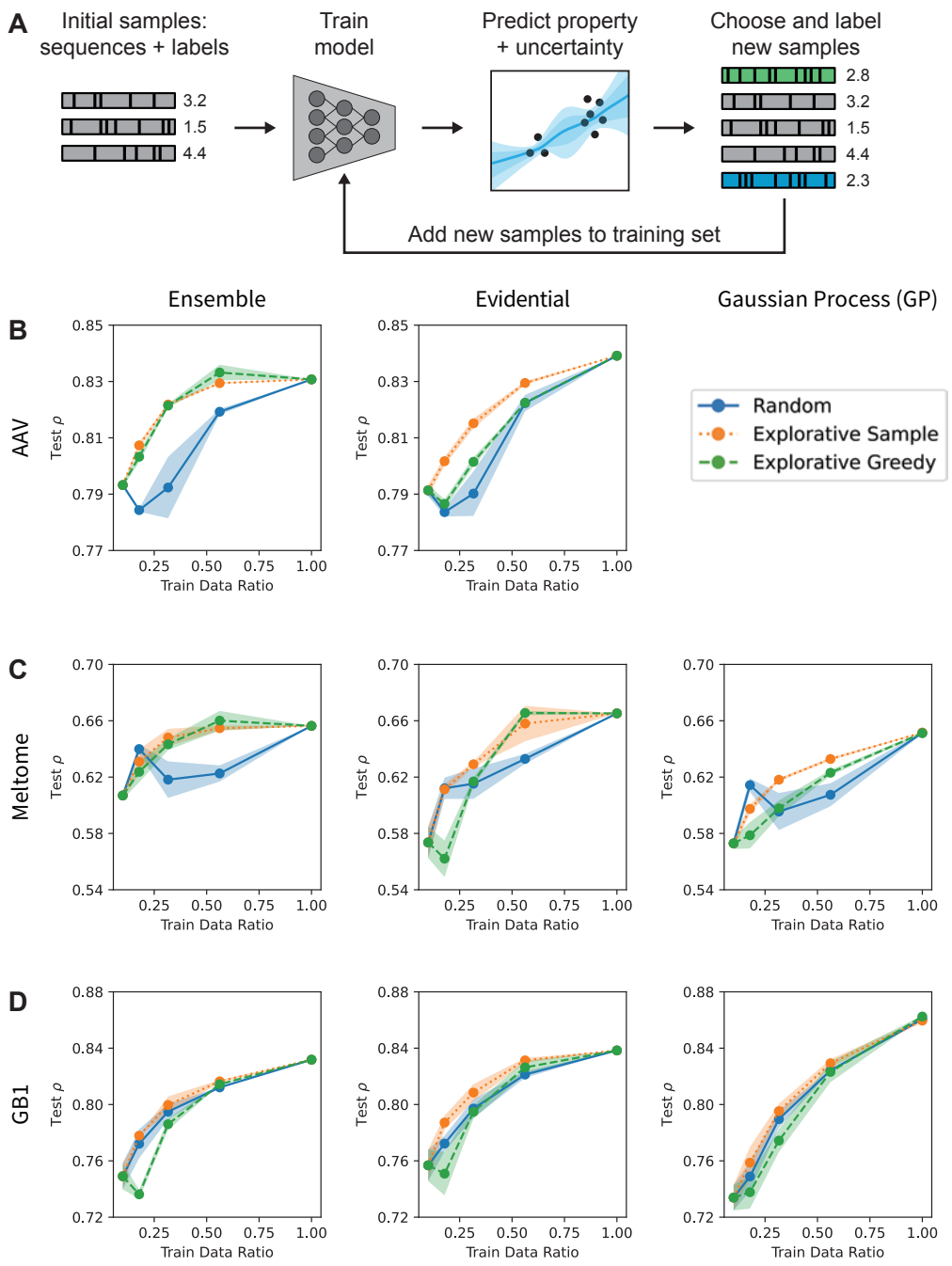

Figure 5: (A) Schematic of active learning approach. A model is trained on an initial dataset, and is then retrained in each iteration by adding more points to the training set based on some selection criteria. (B-D) Uncertainty-guided active learning in protein sequence-function prediction. Spearman rank correlation of predictions (ρ) for the CNN ensemble, CNN evidential, and GP methods evaluated on the AAV/Random (B), Meltome/Random (C), and GB1/Random (D) splits. The "random" strategy acquired sequences with all unseen points having equal probabilities, the "explorative sample" strategy acquired sequences with random sampling weighted by uncertainty, and the "explorative greedy" strategy acquired the previously unseen sequences with the highest uncertainty.

3 Conclusions

Calibrated uncertainty estimations for ML predictions of biomolecular properties are necessary for effective model improvement using active learning or property optimization using Bayesian methods. In this work, we benchmarked a panel of uncertainty quantification (UQ) methods on protein datasets, including on train-test splits that are representative of real-world data collection practices. After evaluating each method based on accuracy, calibration, coverage, width, rank correlation, and performance in active learning, we find that there is no method that performs consistently well across all metrics or all landscapes and splits.

We also examined how models trained using one-hot-encoding representations of sequences compare to those trained on more informative and generalizable representations such as embeddings from a pretrained ESM language model. This comparison illustrated that while the pretrained embeddings do improve model accuracy and uncertainty correlation/calibration in some cases, particularly on splits with higher domain shifts, this is not universally true and in some cases makes performance worse.

While the UQ evaluation metrics used in this work provide valuable information, they are ultimately only a proxy for expected performance in Bayesian optimization and active learning. We found that UQ evaluation metrics are not well-correlated with gains in accuracy from one active learning iteration to another on these datasets. This suggests that future work in UQ should include retrospective Bayesian optimization and/or active learning studies rather than relying on UQ evaluation metrics alone. Our retrospective active learning studies using holdouts of the training sets demonstrate that many of the uncertainty methods outperform random sampling baselines. In some of our experiments, we observe that the uncertainty-based sampling strategies perform worse than random sampling during the earliest stages of active learning, then perform better as a model's accuracy and quality of uncertainty estimates improve in later stages.

Future work in this area could expand on methods (e.g. Bayesian neural networks (*24*) and conformal prediction (*25*)), metrics (e.g. sharpness (*5*), dispersion (*26*), and tightness

(*27*)), and representations (e.g. ESM-2 (*28*) or using an attention layer rather than mean aggregation on our ESM-1b embeddings). While this work considered uncertainty predictions as directly output by the models, further study is needed to understand the effects of post-hoc calibration methods (e.g. scalar recalibration (*26*) or CRUDE (*29*)). Future work should consider additional active learning strategies beyond "explorative greedy" and "explorative sample", such as Thompson sampling (*30*), other exploitative strategies, strategies that consider batch diversity in the acquisition function (*31*), and methods that consider the desired domain shift. Ultimately, this work contributes to a more thorough understanding of how to best apply UQ to sequence-function models and provides a foundation for future work to enable more effective protein engineering.

4 Methods

4.1 Regression Tasks

All tasks studied in this work are regression problems, in which we attempt to fit a model to a dataset with $\mathcal{D}$ data points (x_i, y_i). x_i is a protein sequence representation (either a one-hot encoding or an embedding vector from an ESM language model), and $y_i \in \mathbb{R}$ is a scalar-valued target property from the protein landscapes described in Section 4.2.

4.2 Datasets

The landscapes and splits in this work are taken from the FLIP benchmark (*3*). GB1 is a landscape commonly used for investigating epistasis (interactions between mutations) using the binding domain of protein G, an immunoglobulin binding protein in Streptococcal bacteria. These splits are designed primarily to test generalization from few- to many-mutation sequences. The AAV landscape is based on data collected for the Adeno-associated virus capsid protein, which help the virus integrate a DNA payload into a target cell. The mutations in this landscape are restricted to a subset of positions within a much longer

sequence. The Meltome landscape includes data from proteins across 13 different species for a non-protein-specific property (thermostability), so it includes both local and global variations. The total number of data points in the GB1, AAV, and Meltome sets are 8,733, 284,009, and 27,951, respectively. In the AAV set, 82,583 are sampled (mutations) and 201,426 are designed. For AAV, only the 82,583 sampled sequences are used for the Random and 7 vs. Rest tasks, while all 284,009 are used for the Sampled vs. Designed task.

The names of several of the tasks were changed slightly from the original FLIP nomenclature for clarity: GB1/Random was originally called GB1/Sampled, AAV/Random was originally called AAV/Sampled, AAV/7 vs. Rest was originally called AAV/7 vs. Many, AAV/Sampled vs. Designed was originally called AAV/Mut-Des, and Meltome/Random was originally called Meltome/Mixed.

4.3 ESM Embeddings

We used the pretrained, 650M-parameter ESM-1b model (`esm1b_t33_650M_UR50S`) from (*15*) to generate embeddings of the protein sequences in this study and to compare these embeddings to one-hot encoding representations. Sequence embeddings from the final representation layer (layer 33) were mean pooled per amino acid over the length of each protein sequence, which resulted in a fixed embedding size of 1280 for each sequence.

4.4 Base CNN Model Architectures

The base architecture of all CNN models in this work was taken from the CNNs in the FLIP benchmark (*3*). For the one-hot encoding inputs, this was comprised of a convolution with 1024 channels and kernel width 5, a ReLU non-linear activation function, a linear mapping to 2048 dimensions, a max pool over the sequence, and a linear mapping to 1 dimension. For ESM embedding inputs, the architecture was the same except with 1280 input channels rather than 1024, and a linear mapping to 2560 dimensions rather than 2048.

4.5 CNN Model Training Procedures

To train our CNN models, we used a batch size of 256 (GB1, AAV) or 30 (Meltome). Adam (*32*) was used for optimization with the following learning rates: 0.001 for the convolution weights, 0.00005 for the first linear mapping, and 0.000005 for the second linear mapping. Weight decay was set to 0.05 for both the first and second linear mappings. CNNs were trained with early stopping using a patience of 20 epochs. Each model was trained on an NVIDIA Volta V100 GPU. Code, data, and instructions needed to reproduce results can be found at `https://github.com/microsoft/protein-uq`.

4.6 Uncertainty Methods

For all models and landscapes, the sequences were featurized using either one-hot encodings or embeddings from a pretrained language model (see Section 4.3).

We used the `scikit-learn` (*33*) implementation of Bayesian ridge regression (BRR) with default hyperparameters. BRR for one-hot encodings of the Meltome/Random split was not feasible because the required work array was too large to perform the computation with standard 32-bit LAPACK in `scipy`.

For Gaussian processes (GPs), we used the GPyTorch (*34*) implementation with the constant mean module, scaled rational quadratic (RQ) kernel covariance module, and Gaussian likelihood. Some GP models (for AAV one-hot encodings and ESM embeddings, and Meltome one-hot encodings) were not feasible to train due to GPU-memory requirements for exact GP models, so these are omitted from the results.

For our uncertainty methods that rely on sampling (dropout, ensemble, and SVI), the final model prediction is defined as the mean of the set of inference samples, and the uncertainty is the standard deviation of these samples. In other words, for a set of predictions $\mathcal{E} = \{G_1(x), G_2(x), ..., G_n(x)\}$ (each coming from an individual model G_i), the final prediction is defined as

$$\hat{G}(x) = \sum_{G \in \mathcal{E}} \frac{G(x)}{n}$$

and the uncertainty $U(x)$ is defined as

$$U(x) = \sqrt{\sum_{G \in \mathcal{E}} \frac{(\hat{G}(x) - G(x))^2}{n}}$$

The uncertainty is sometimes defined as the variance U^2, but using the standard deviation puts the uncertainty in the same units as the predictions.

For dropout uncertainty (*19*), a single model G was trained normally. At inference time, we applied $n = 10$ random dropout masks with dropout probability p to obtain the set of predictions $\mathcal{E}$ for each input x_i. We tested dropout rates of $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and reported the model with the lowest miscalibration area.

Similarly for last-layer stochastic variational inference (SVI) (*23*), we obtained $\mathcal{E}$ using $n = 10$ samples from a set of models where each G_i has the weight and bias terms of its last layer themselves sampled from a distribution $q(\theta)$ that has been trained to approximate the true posterior $p(\theta|\mathcal{D})$.

Traditional model ensembling calculated $\mathcal{E}$ using $n = 5$ models trained using different random seeds for initialization of the CNN parameters and batching / stochastic gradient descent. The computational cost of this approach is 5 times that of a standard CNN model since the cost scales linearly with the size of the ensemble.

In mean-variance estimation (MVE) models, we adapt the base CNN architecture to produce 2 outputs ($\theta = \{\mu, \sigma^2\}$) for each data point (x_i, y_i) in the last layer rather than 1, and we train using the negative log-likelihood loss:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{(y_i - \mu(x_i))^2}{2\sigma^2(x_i)} + \frac{1}{2} \log(2\pi\sigma^2(x_i))$$

In practice, the variance (σ^2) is clamped to a minimum value of 10^{-6} to prevent division by

0.

Evidential deep learning modifies the loss function of the traditional CNN to jointly maximize the model's fit to data while also minimizing its evidence on errors (increasing uncertainty on unreliable predictions) (*21*):

$$\mathcal{L}(x) = \mathcal{L}^{NLL}(x) + \lambda \mathcal{L}^{R}(x)$$

where $\mathcal{L}^{NLL}(x)$ is the negative log-likelihood, $\mathcal{L}^{R}(x)$ is the evidence regularizer, and λ controls the trade-off between these two terms. In this study, we use $\lambda = 1$ for all evidential models. In these models, the last layer of the model produces 4 outputs $\mathbf{m} = \{\gamma, \nu, \alpha, \beta\}$ that parameterize the Normal-Inverse-Gamma distribution. This distribution assumes that targets y_i are drawn i.i.d. from a Gaussian distribution with unknown mean and variance $\theta = \{\mu, \sigma^2\}$, where the mean is drawn from a Gaussian and the variance is drawn from an Inverse-Gamma distribution. The output of the evidential model can be divided into the prediction and the epistemic (model) and aleatoric (data) uncertainty components following the analysis of Amini et al. (*21*):

$$\underbrace{\mathbb{E}[\mu] = \gamma}_{\text{prediction}}, \quad \underbrace{\mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}}_{\text{aleatoric}}, \quad \underbrace{\text{Var}[\mu] = \frac{\beta}{\nu(\alpha - 1)}}_{\text{epistemic}}$$

We report the sum of the aleatoric and epistemic uncertainties as the total uncertainty.

4.7 Evaluation Metrics

To give a comprehensive report of model accuracy, we computed the following metrics on the test sets: root mean square error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), and Spearman rank correlation (ρ). RMSE is more sensitive to outliers than MAE, so while both are informative independently, the combination of the two gives additional information about the distribution of errors. R^2 and ρ are both unitless and are

thus more easily interpreted and compared across datasets.

We evaluated the quality of the uncertainty estimates using four metrics. First, ρ_{unc} is the Spearman rank correlation between uncertainty and absolute prediction error. Following Kompa et al. (*35*), we measured the coverage as the percentage of true values that fall within the 95% confidence interval ($\pm 2\sigma$) of each prediction. Kompa et al. (*35*) define the width as the size of the 95% confidence region (4σ), but we normalized this width relative to the range (R) of the training set as $4\sigma/R$ to make these values unitless and thus more interpretable across datasets. Finally, the miscalibration area (also called the area under the calibration error curve or AUCE) quantifies the absolute difference between the calibration plot and perfect calibration in a single number (*36*).

4.8 Active Learning

Each active learning run began with a random sample of 10% of the full training data. We evaluated several alternatives for adding to this initial dataset using different sampling strategies (acquisition functions): explorative greedy, explorative sample, and random. "Explorative greedy" sampled the sequences with the highest uncertainty; "explorative sample" sampled the data according to the probability of sampling a sequence equal to the ratio of its uncertainty to the sum of all uncertainties in the dataset (i.e. random sampling weighted by uncertainty); and "random" sampled the data uniformly from all unobserved sequences. We employed these sampling strategies 5 times in each active learning run, with the 5 training set sizes equally spaced on a log scale. We repeated this process using 3 folds (different random seeds for sampling initial dataset and "explorative sample" probabilities) and calculated the mean and standard deviation across these folds.

Acknowledgement

K.P.G. was supported by a Microsoft Research micro-internship and by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1745302. The authors thank the MIT Lincoln Laboratory Supercloud cluster (*37*) at the Massachusetts Green High Performance Computing Center (MGHPCC) for providing high-performance computing resources to train our machine learning models.

5 Author Contributions

A.P.A. and K.K.Y. conceived the project. K. P. G. wrote the computer code, analyzed the data, and wrote the first manuscript draft. A.P.A. and K.K.Y. supervised the research and edited the manuscript.

Supporting Information Available

Code availability, OHE results, OHE vs. ESM results comparison, additional prediction and uncertainty evaluation metrics, and additional active learning results.

References

1. Yang, K. K., Wu, Z., and Arnold, F. H. (2019) Machine-learning-guided directed evolution for protein engineering. *Nature methods 16*, 687–694.

2. Kendall, A., and Gal, Y. (2017) What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems 30*.

3. Dallago, C., Mou, J., Johnston, K. E., Wittmann, B., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for

proteins. Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). 2021.

4. Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., and Green, W. H. (2020) Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of chemical information and modeling 60*, 2697–2717.

5. Tran, K., Neiswanger, W., Yoon, J., Zhang, Q., Xing, E., and Ulissi, Z. W. (2020) Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology 1*, 025006.

6. Hirschfeld, L., Swanson, K., Yang, K., Barzilay, R., and Coley, C. W. (2020) Uncertainty quantification using neural networks for molecular property prediction. *Journal of Chemical Information and Modeling 60*, 3770–3780.

7. Nigam, A., Pollice, R., Hurley, M. F., Hickman, R. J., Aldeghi, M., Yoshikawa, N., Chithrananda, S., Voelz, V. A., and Aspuru-Guzik, A. (2021) Assigning confidence to molecular property prediction. *Expert opinion on drug discovery 16*, 1009–1023.

8. Soleimany, A. P., Amini, A., Goldman, S., Rus, D., Bhatia, S. N., and Coley, C. W. (2021) Evidential deep learning for guided molecular property prediction and discovery. *ACS central science 7*, 1356–1367.

9. Gruich, C., Madhavan, V., Wang, Y., and Goldsmith, B. (2023) Clarifying Trust of Materials Property Predictions using Neural Networks with Distribution-Specific Uncertainty Quantification. *arXiv preprint arXiv:2302.02595*

10. Mariet, Z., Jerfel, G., Wang, Z., Angermüller, C., Belanger, D., Vora, S., Bileschi, M., Colwell, L., Sculley, D., Tran, D., et al. Deep Uncertainty and the Search for Proteins. Workshop: Machine Learning for Molecules. 2020.

11. Hie, B., Bryson, B. D., and Berger, B. (2020) Leveraging Uncertainty in Machine Learning Accelerates Biological Discovery and Design. *Cell Systems 11*, 461–477.e9.

12. Parkinson, J., and Wang, W. (2023) Scalable Gaussian process regression enables accurate prediction of protein and small molecule properties with uncertainty quantitation. *arXiv preprint arXiv:2302.03294*

13. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*

14. Gruver, N., Stanton, S., Kirichenko, P., Finzi, M., Maffettone, P., Myers, V., Delaney, E., Greenside, P., and Wilson, A. G. Effective surrogate models for protein design with bayesian optimization. ICML Workshop on Computational Biology. 2021.

15. Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences 118*, e2016239118.

16. MacKay, D. J. (1992) Bayesian interpolation. *Neural computation 4*, 415–447.

17. Tipping, M. E. (2001) Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research 1*, 211–244.

18. Rasmussen, C. E., and Williams, C. Gaussian processes for machine learning, vol. 1. 2006.

19. Gal, Y., and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. Proceedings of The 33rd International Conference on Machine Learning. New York, New York, USA, 2016; pp 1050–1059.

20. Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. Advances in Neural Information Processing Systems. 2017.

21. Amini, A., Schwarting, W., Soleimany, A., and Rus, D. Deep Evidential Regression. Advances in Neural Information Processing Systems. 2020; pp 14927–14937.

22. Nix, D. A., and Weigend, A. S. Estimating the mean and variance of the target probability distribution. Proceedings of 1994 ieee international conference on neural networks (ICNN'94). 1994; pp 55–60.

23. Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013) Stochastic variational inference. *Journal of Machine Learning Research*

24. Neal, R. M. *Bayesian learning for neural networks*; Springer Science & Business Media, 2012; Vol. 118.

25. Norinder, U., Carlsson, L., Boyer, S., and Eklund, M. (2014) Introducing conformal prediction in predictive modeling. A transparent and flexible alternative to applicability domain determination. *Journal of chemical information and modeling 54*, 1596–1603.

26. Levi, D., Gispan, L., Giladi, N., and Fetaya, E. (2022) Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors 22*, 5540.

27. Gneiting, T., and Raftery, A. E. (2007) Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association 102*, 359–378.

28. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. (2022) Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*

29. Zelikman, E., Healy, C., Zhou, S., and Avati, A. (2020) CRUDE: calibrating regression uncertainty distributions empirically. *arXiv preprint arXiv:2005.12496*

30. Chapelle, O., and Li, L. (2011) An empirical evaluation of thompson sampling. *Advances in neural information processing systems 24*.

31. Kirsch, A., Van Amersfoort, J., and Gal, Y. (2019) BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems 32*.

32. Kingma, D. P., and Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*

33. Pedregosa, F. et al. (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

34. Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018) Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems 31*.

35. Kompa, B., Snoek, J., and Beam, A. L. (2021) Empirical Frequentist Coverage of Deep Learning Uncertainty Quantification Procedures. *Entropy 23*.

36. Gustafsson, F. K., Danelljan, M., and Schon, T. B. Evaluating scalable bayesian deep learning methods for robust computer vision. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020; pp 318–319.

37. Reuther, A. et al. Interactive supercomputing on 40,000 cores for machine learning and data analysis. 2018 IEEE High Performance extreme Computing Conference (HPEC). 2018; pp 1–6.

Supporting Information

Benchmarking Uncertainty Quantification for Protein Engineering

Kevin P. Greenman,[†] Ava P. Amini,[*,‡] and Kevin K. Yang[*,‡]

[†]*Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA*

[‡]*Microsoft Research New England, Cambridge, MA, USA*

E-mail: ava.amini@microsoft.com; yang.kevin@microsoft.com

1 Code Availability

The code for the models, uncertainty methods, and evaluation metrics in this work is available at `https://github.com/microsoft/protein-uq`.

2 OHE Results

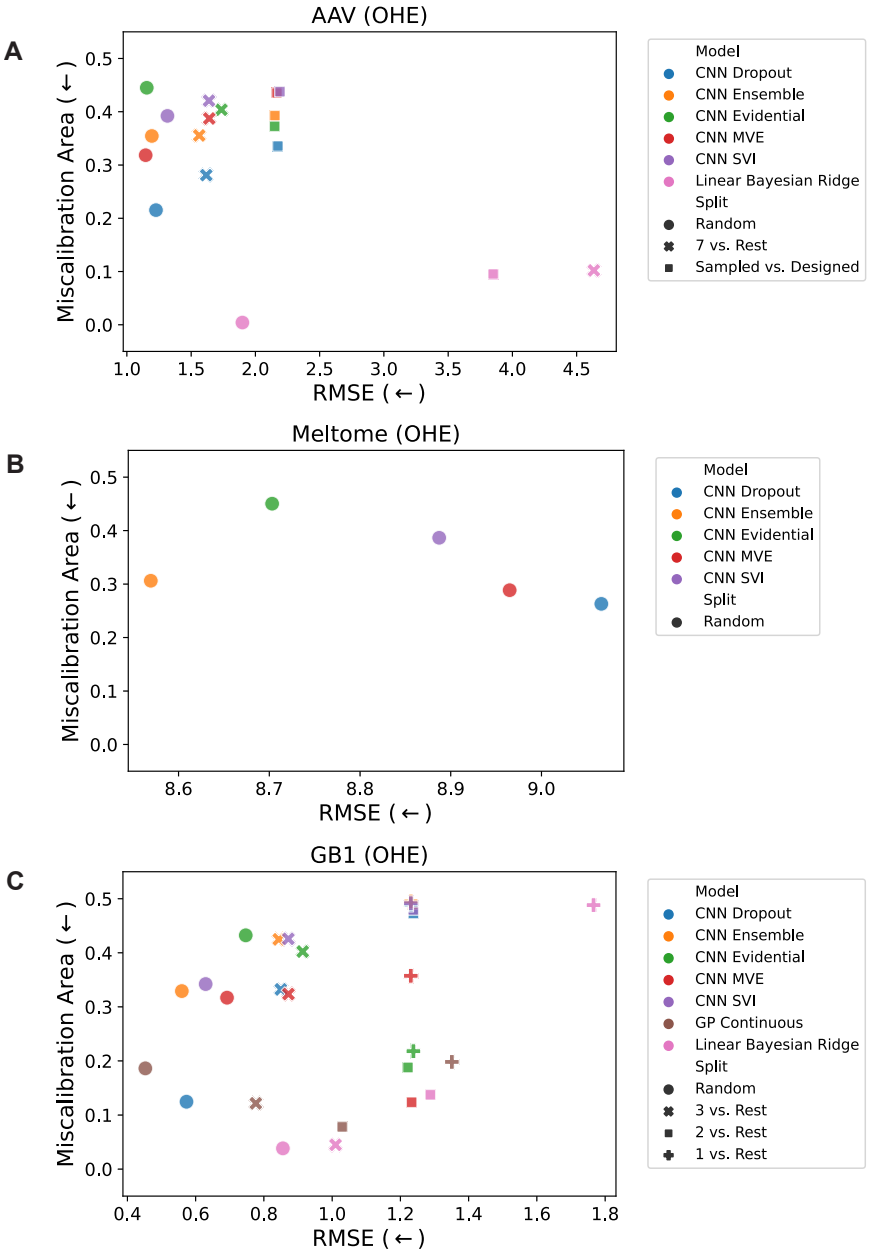

Figure S1: Miscalibration area vs. root mean square error (RMSE) for the (a) AAV, (b) Meltome, and (c) GB1 landscapes. Miscalibration area (also called the area under the calibration error curve or AUCE) quantifies the absolute difference between the calibration plot and perfect calibration. It is desirable to have a model that is both accurate and well-calibrated, so the best performing points are those closest to the lower left corner of the plots. The GP Continuous model is not shown for the AAV landscape due to memory constraints for training these models. The GP Continuous and Linear Bayesian Ridge models are not shown for the Meltome landscape due to memory constraints and limitations of 32-bit LAPACK, respectively.

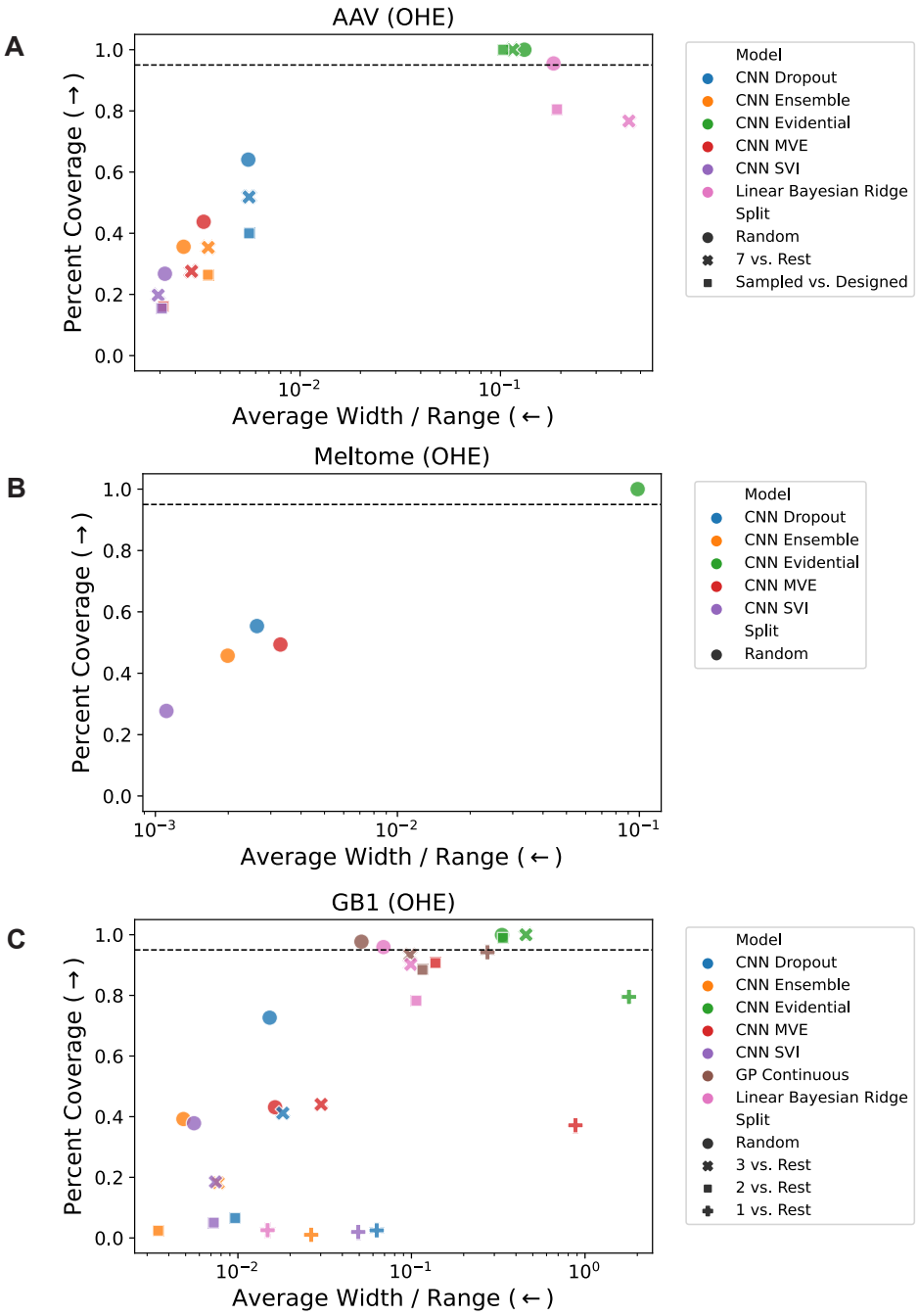

Figure S2: Coverage vs. average width / range for the (a) AAV, (b) Meltome, and (c) GB1 landscapes. Coverage is the percentage of true values that fall within the 95% confidence interval ($\pm 2\sigma$) of each prediction, and the width is the size of the 95% confidence region relative to the range of the training set ($4\sigma/R$ where R is the range of the training set). A good model exhibits high coverage and low width, which corresponds to the upper left of each plot. The horizontal dashed line indicates 95% coverage. The GP Continuous model is not shown for the AAV landscape due to memory constraints for training these models. The GP Continuous and Linear Bayesian Ridge models are not shown for the Meltome landscape due to memory constraints and limitations of 32-bit LAPACK, respectively.

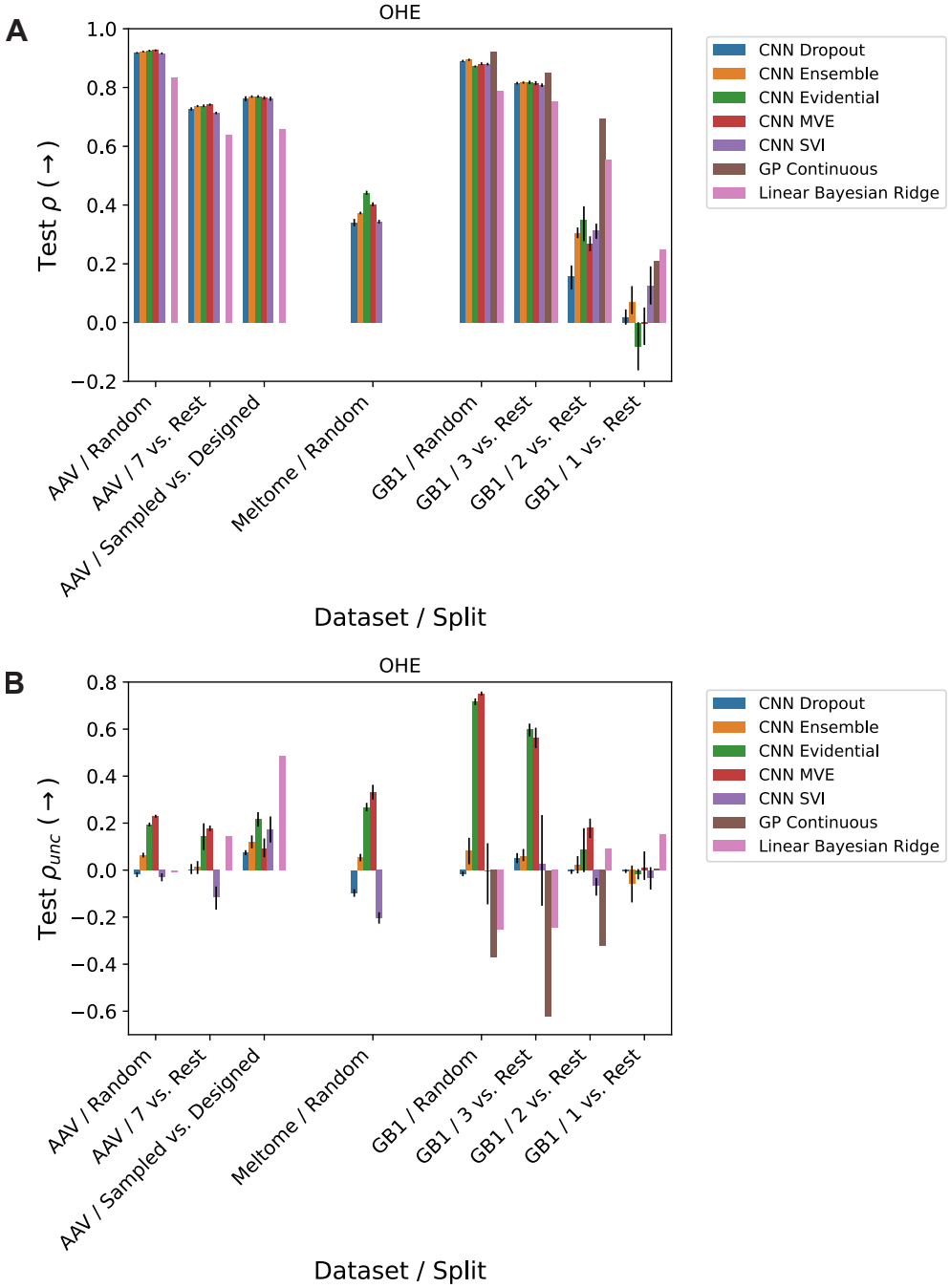

Figure S3: Spearman rank correlations of (a) predictions (ρ) and (b) uncertainties (ρ_{unc}) vs. extrapolation. Within each landscape (AAV, Meltome, and GB1), splits are qualitatively ordered by the amount of domain shift between train and test sets, with the lowest domain shift on the left and the highest domain shift on the right. Error bars on the CNN results represent the 95% confidence interval calculated from 5 different random initializations of the CNN parameters.

3 OHE vs. ESM Comparison

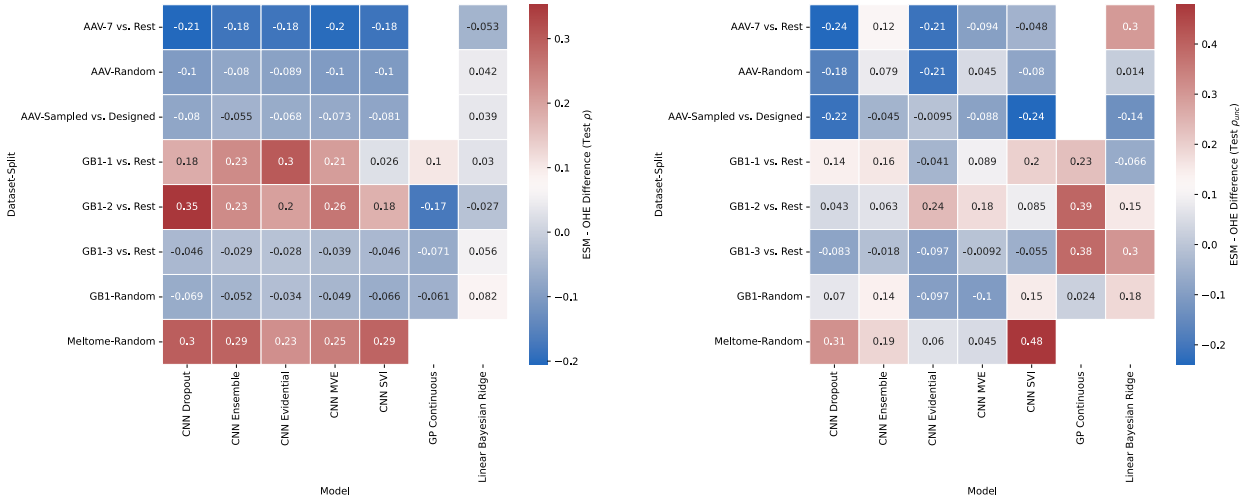

Figure S4: Comparison of prediction (ρ) and uncertainty (ρ_{unc}) performance between the OHE and ESM representations across all models and tasks. Red cells indicate that the ESM representation performed better, while blue cells indicate that the OHE representation performed better.

4 Prediction and Uncertainty Evaluation Metrics

Table S1: Test set RMSE for models trained on OHE representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	1.618	1.563	1.736	1.641	1.639	NaN	4.634
	Random	1.226	1.194	1.155	1.146	1.316	NaN	1.899
	Sampled vs. Designed	2.175	2.152	2.150	2.168	2.191	NaN	3.850
GB1	1 vs. Rest	1.231	1.231	1.238	1.231	1.231	1.351	1.766
	2 vs. Rest	1.238	1.237	1.222	1.233	1.238	1.029	1.288
	3 vs. Rest	0.850	0.844	0.914	0.872	0.872	0.776	1.010
	Random	0.573	0.559	0.747	0.692	0.629	0.453	0.856
Meltome	Random	9.066	8.569	8.703	8.965	8.887	NaN	NaN

Table S2: Test set MAE for models trained on OHE representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	1.274	1.229	1.350	1.268	1.311	NaN	3.364
	Random	0.937	0.909	0.863	0.863	1.017	NaN	1.499
	Sampled vs. Designed	1.720	1.682	1.670	1.703	1.732	NaN	2.676
GB1	1 vs. Rest	0.908	0.909	0.903	0.909	0.909	1.161	1.472
	2 vs. Rest	0.990	0.986	0.913	0.977	0.984	0.675	1.002
	3 vs. Rest	0.617	0.609	0.599	0.592	0.630	0.511	0.780
	Random	0.400	0.382	0.461	0.431	0.438	0.302	0.648
Meltome	Random	7.019	6.597	6.539	6.795	6.845	NaN	NaN

Table S3: Test set R^2 for models trained on OHE representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.366	0.409	0.271	0.348	0.347	NaN	-4.195
	Random	0.841	0.850	0.859	0.861	0.816	NaN	0.620
	Sampled vs. Designed	0.608	0.616	0.617	0.610	0.602	NaN	-0.229
GB1	1 vs. Rest	-0.016	-0.015	-0.027	-0.015	-0.015	-0.223	-1.090
	2 vs. Rest	-0.012	-0.010	0.013	-0.004	-0.012	0.300	-0.095
	3 vs. Rest	0.556	0.562	0.487	0.532	0.533	0.629	0.373
	Random	0.774	0.785	0.616	0.670	0.727	0.859	0.496
Meltome	Random	0.391	0.456	0.439	0.405	0.415	NaN	NaN

Table S4: Test set ρ for models trained on OHE representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.727	0.737	0.738	0.742	0.713	NaN	0.640
	Random	0.918	0.922	0.926	0.927	0.916	NaN	0.834
	Sampled vs. Designed	0.762	0.769	0.770	0.765	0.762	NaN	0.657
GB1	1 vs. Rest	0.018	0.071	-0.081	-0.006	0.126	0.211	0.249
	2 vs. Rest	0.156	0.304	0.351	0.268	0.313	0.694	0.555
	3 vs. Rest	0.815	0.817	0.818	0.815	0.808	0.850	0.753
	Random	0.890	0.894	0.872	0.881	0.880	0.922	0.789
Meltome	Random	0.340	0.373	0.441	0.403	0.344	NaN	NaN

Table S5: Test set ρ_{unc} for models trained on OHE representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.001	0.013	0.142	0.178	-0.114	NaN	0.145
	Random	-0.019	0.064	0.194	0.229	-0.030	NaN	-0.007
	Sampled vs. Designed	0.074	0.120	0.216	0.093	0.173	NaN	0.486
GB1	1 vs. Rest	-0.005	-0.059	-0.015	0.009	-0.032	0.004	0.153
	2 vs. Rest	-0.006	0.022	0.088	0.181	-0.067	-0.323	0.092
	3 vs. Rest	0.050	0.060	0.598	0.563	0.028	-0.621	-0.246
	Random	-0.016	0.085	0.718	0.752	-0.003	-0.370	-0.254
Meltome	Random	-0.097	0.053	0.267	0.331	-0.203	NaN	NaN

Table S6: Test set % coverage for models trained on OHE representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.519	0.354	1.000	0.276	0.198	NaN	0.767
	Random	0.641	0.356	1.000	0.438	0.268	NaN	0.955
	Sampled vs. Designed	0.400	0.264	1.000	0.161	0.154	NaN	0.805
GB1	1 vs. Rest	0.025	0.010	0.795	0.372	0.020	0.942	0.026
	2 vs. Rest	0.066	0.024	0.990	0.907	0.050	0.884	0.783
	3 vs. Rest	0.412	0.182	1.000	0.440	0.185	0.933	0.902
	Random	0.727	0.392	0.999	0.431	0.378	0.977	0.959
Meltome	Random	0.554	0.457	1.000	0.494	0.277	NaN	NaN

Table S7: Test set $4\sigma/R$ for models trained on OHE representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.006	0.003	0.115	0.003	0.002	NaN	0.438
	Random	0.006	0.003	0.132	0.003	0.002	NaN	0.184
	Sampled vs. Designed	0.006	0.003	0.103	0.002	0.002	NaN	0.191
GB1	1 vs. Rest	0.063	0.026	1.791	0.881	0.049	0.274	0.015
	2 vs. Rest	0.010	0.003	0.336	0.137	0.007	0.116	0.107
	3 vs. Rest	0.018	0.008	0.457	0.030	0.007	0.098	0.099
	Random	0.015	0.005	0.332	0.016	0.006	0.052	0.069
Meltome	Random	0.003	0.002	0.098	0.003	0.001	NaN	NaN

Table S8: Test set miscalibration area for models trained on OHE representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.281	0.355	0.404	0.387	0.421	NaN	0.102
	Random	0.215	0.354	0.445	0.318	0.392	NaN	0.004
	Sampled vs. Designed	0.336	0.393	0.373	0.435	0.438	NaN	0.095
GB1	1 vs. Rest	0.489	0.495	0.218	0.357	0.492	0.198	0.488
	2 vs. Rest	0.473	0.490	0.188	0.124	0.479	0.078	0.137
	3 vs. Rest	0.332	0.425	0.402	0.324	0.426	0.122	0.045
	Random	0.125	0.329	0.432	0.317	0.342	0.186	0.038
Meltome	Random	0.263	0.306	0.450	0.289	0.387	NaN	NaN

Table S9: Test set $\overline{NLL}$ for models trained on OHE representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	5.427	18.148	3.342	26.312	51.394	NaN	3.667
	Random	3.678	19.773	3.430	7.856	24.652	NaN	2.058
	Sampled vs. Designed	9.132	31.511	3.266	64.200	62.343	NaN	3.183
GB1	1 vs. Rest	2977.270	23918.764	3.259	12.176	3946.655	1.937	9871.345
	2 vs. Rest	251.599	2695.786	1.834	1.790	380.500	1.596	2.076
	3 vs. Rest	12.551	96.398	2.294	56.250	74.424	1.216	1.488
	Random	2.413	29.654	2.165	90.282	24.882	0.757	1.265
Meltome	Random	7.764	11.874	5.580	28.920	27.409	NaN	NaN

Table S10: Test set $\overline{NLL_{opt}}$ for models trained on OHE representation

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	1.237	1.202	1.282	1.214	1.277	NaN	2.183
	Random	0.899	0.868	0.797	0.805	0.991	NaN	1.408
	Sampled vs. Designed	1.538	1.498	1.481	1.517	1.542	NaN	1.834
GB1	1 vs. Rest	0.878	0.879	0.855	0.881	0.883	1.293	1.483
	2 vs. Rest	1.055	1.049	0.908	1.033	1.042	0.418	0.977
	3 vs. Rest	0.438	0.414	0.269	0.308	0.433	0.037	0.730
	Random	-0.079	-0.180	-0.223	-0.451	0.025	-0.374	0.541
Meltome	Random	2.929	2.851	2.820	2.875	2.891	NaN	NaN

Table S11: Test set $\overline{NLL}$ / $\overline{NLL_{opt}}$ ratio for models trained on OHE representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	4.387	15.102	2.608	21.684	38.925	NaN	1.680
	Random	4.089	22.765	4.305	9.750	24.970	NaN	1.462
	Sampled vs. Designed	5.939	21.014	2.206	42.262	40.329	NaN	1.736
GB1	1 vs. Rest	3390.230	27281.195	3.810	13.818	4474.286	1.498	6656.782
	2 vs. Rest	238.554	2564.788	2.024	1.732	365.159	3.816	2.125
	3 vs. Rest	28.691	232.149	8.737	175.004	173.197	33.170	2.038
	Random	-49.594	-161.397	-10.754	-204.256	220.658	-2.026	2.339
Meltome	Random	2.652	4.164	1.979	10.050	9.483	NaN	NaN

Table S12: Test set RMSE for models trained on ESM representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	2.241	2.213	2.157	2.168	2.217	NaN	4.083
	Random	1.875	1.734	1.719	1.793	1.855	NaN	1.582
	Sampled vs. Designed	2.648	2.555	2.581	2.639	2.649	NaN	2.942
GB1	1 vs. Rest	1.333	1.314	1.306	1.328	1.383	1.360	2.413
	2 vs. Rest	1.127	1.111	1.086	1.118	1.140	1.111	1.484
	3 vs. Rest	0.918	0.885	0.908	0.911	0.930	0.920	0.949
	Random	0.761	0.719	0.812	0.775	0.780	0.669	0.631
Meltome	Random	6.722	6.536	6.598	6.764	6.713	6.447	6.936

Table S13: Test set MAE for models trained on ESM representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	1.844	1.822	1.738	1.769	1.819	NaN	2.798
	Random	1.474	1.354	1.306	1.400	1.458	NaN	1.235
	Sampled vs. Designed	2.218	2.140	2.137	2.199	2.203	NaN	2.044
GB1	1 vs. Rest	1.143	1.127	1.110	1.137	1.188	1.173	1.949
	2 vs. Rest	0.868	0.858	0.819	0.862	0.886	0.845	1.123
	3 vs. Rest	0.687	0.656	0.627	0.654	0.697	0.680	0.731
	Random	0.537	0.502	0.513	0.515	0.556	0.471	0.478
Meltome	Random	5.024	4.862	4.806	4.974	4.999	4.777	5.218

Table S14: Test set R^2 for models trained on ESM representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	-0.216	-0.187	-0.128	-0.140	-0.193	NaN	-3.034
	Random	0.628	0.683	0.687	0.659	0.636	NaN	0.736
	Sampled vs. Designed	0.418	0.459	0.447	0.422	0.418	NaN	0.282
GB1	1 vs. Rest	-0.191	-0.157	-0.144	-0.183	-0.284	-0.240	-2.900
	2 vs. Rest	0.162	0.185	0.222	0.174	0.142	0.185	-0.455
	3 vs. Rest	0.481	0.518	0.493	0.489	0.467	0.480	0.446
	Random	0.600	0.644	0.546	0.585	0.580	0.692	0.726
Meltome	Random	0.665	0.684	0.678	0.661	0.666	0.692	0.644

Table S15: Test set ρ for models trained on ESM representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.522	0.557	0.557	0.542	0.531	NaN	0.586
	Random	0.816	0.842	0.837	0.826	0.813	NaN	0.876
	Sampled vs. Designed	0.682	0.714	0.701	0.692	0.681	NaN	0.696
GB1	1 vs. Rest	0.202	0.303	0.222	0.200	0.152	0.315	0.279
	2 vs. Rest	0.509	0.535	0.552	0.530	0.490	0.523	0.528
	3 vs. Rest	0.769	0.788	0.790	0.776	0.762	0.779	0.809
	Random	0.822	0.842	0.837	0.832	0.813	0.861	0.871
Meltome	Random	0.638	0.664	0.668	0.654	0.633	0.650	0.591

Table S16: Test set ρ_{unc} for models trained on ESM representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	-0.235	0.129	-0.069	0.084	-0.162	NaN	0.441
	Random	-0.194	0.143	-0.016	0.274	-0.110	NaN	0.008
	Sampled vs. Designed	-0.142	0.075	0.206	0.005	-0.066	NaN	0.349
GB1	1 vs. Rest	0.136	0.099	-0.057	0.099	0.166	0.233	0.087
	2 vs. Rest	0.037	0.084	0.323	0.358	0.018	0.068	0.245
	3 vs. Rest	-0.033	0.043	0.501	0.554	-0.027	-0.237	0.056
	Random	0.054	0.223	0.621	0.647	0.145	-0.347	-0.072
Meltome	Random	0.214	0.241	0.327	0.376	0.275	0.109	0.119

Table S17: Test set % coverage for models trained on ESM representation (↑)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.082	0.289	1.000	0.499	0.052	NaN	0.721
	Random	0.110	0.352	1.000	0.759	0.047	NaN	0.950
	Sampled vs. Designed	0.064	0.195	1.000	0.378	0.039	NaN	0.817
GB1	1 vs. Rest	0.104	0.126	0.985	0.639	0.079	0.943	0.012
	2 vs. Rest	0.146	0.183	0.798	0.264	0.099	0.913	0.801
	3 vs. Rest	0.127	0.289	0.998	0.571	0.076	0.935	0.890
	Random	0.161	0.377	1.000	0.586	0.102	0.968	0.947
Meltome	Random	0.097	0.283	1.000	0.711	0.059	0.956	0.932

Table S18: Test set $4\sigma/R$ for models trained on ESM representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.001	0.005	2.989	0.008	0.001	NaN	0.162
	Random	0.001	0.004	0.157	0.011	0.000	NaN	0.155
	Sampled vs. Designed	0.001	0.004	0.425	0.008	0.001	NaN	0.152
GB1	1 vs. Rest	0.054	0.062	0.555	0.254	0.042	0.275	0.008
	2 vs. Rest	0.013	0.018	0.127	0.031	0.009	0.120	0.124
	3 vs. Rest	0.006	0.013	0.271	0.042	0.003	0.102	0.088
	Random	0.002	0.006	0.148	0.019	0.002	0.063	0.050
Meltome	Random	0.000	0.001	0.030	0.003	0.000	0.605	0.580

Table S19: Test set miscalibration area for models trained on ESM representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	0.467	0.387	0.457	0.297	0.479	NaN	0.173
	Random	0.455	0.356	0.392	0.162	0.481	NaN	0.016
	Sampled vs. Designed	0.474	0.423	0.414	0.348	0.484	NaN	0.107
GB1	1 vs. Rest	0.459	0.451	0.053	0.268	0.469	0.202	0.494
	2 vs. Rest	0.442	0.427	0.107	0.397	0.461	0.057	0.121
	3 vs. Rest	0.449	0.379	0.293	0.273	0.469	0.022	0.066
	Random	0.434	0.340	0.345	0.252	0.461	0.123	0.037
Meltome	Random	0.460	0.384	0.383	0.190	0.476	0.055	0.019

Table S20: Test set $\overline{NLL}$ for models trained on ESM representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	453.943	19.534	5.207	10.936	1370.946	NaN	3.667
	Random	320.970	16.868	3.338	2.488	1212.551	NaN	1.877
	Sampled vs. Designed	444.501	44.511	4.052	9.440	1913.293	NaN	2.724
GB1	1 vs. Rest	64.584	77.056	1.710	2.402	99.813	1.948	25929.804
	2 vs. Rest	138.280	76.581	7.136	599.876	293.891	1.595	2.139
	3 vs. Rest	273.534	43.638	1.726	39.261	671.193	1.348	1.468
	Random	247.572	20.924	1.647	19.394	565.041	1.053	0.966
Meltome	Random	301.052	30.996	4.423	4.672	703.508	3.282	3.360

Table S21: Test set $\overline{NLL_{opt}}$ for models trained on ESM representation

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	1.664	1.653	1.567	1.608	1.644	NaN	1.911
	Random	1.367	1.278	1.205	1.310	1.359	NaN	1.188
	Sampled vs. Designed	1.871	1.838	1.812	1.855	1.851	NaN	1.639
GB1	1 vs. Rest	1.275	1.259	1.237	1.268	1.309	1.304	1.694
	2 vs. Rest	0.847	0.841	0.764	0.836	0.883	0.864	1.053
	3 vs. Rest	0.565	0.493	0.384	0.454	0.585	0.543	0.677
	Random	0.221	0.135	-0.042	0.029	0.281	0.083	0.235
Meltome	Random	2.551	2.510	2.464	2.519	2.537	2.477	2.586

Table S22: Test set $\overline{NLL}$ / $\overline{NLL_{opt}}$ ratio for models trained on ESM representation (↓)

Dataset	Model Split	Dropout	Ensemble	Evidential	MVE	SVI	GP	BRR
AAV	7 vs. Rest	273.638	11.806	3.324	6.734	826.919	NaN	1.919
	Random	228.987	13.229	2.786	1.915	868.280	NaN	1.579
	Sampled vs. Designed	235.180	24.210	2.239	5.094	1013.232	NaN	1.662
GB1	1 vs. Rest	50.427	61.282	1.385	1.901	77.453	1.494	15310.669
	2 vs. Rest	160.595	91.310	8.750	697.394	328.277	1.846	2.031
	3 vs. Rest	453.390	89.292	4.715	90.982	1070.519	2.483	2.168
	Random	991.498	168.540	-6.076	42.706	1751.474	12.645	4.110
Meltome	Random	117.634	12.360	1.796	1.856	276.206	1.325	1.299

5 Active Learning

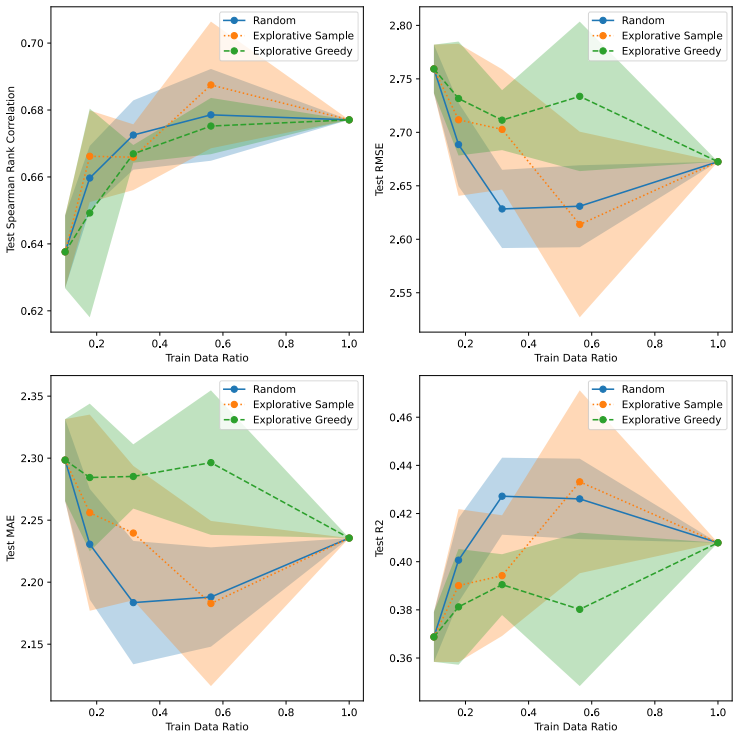

Figure S5: Active learning results for AAV/Sampled vs. Designed using CNN Dropout uncertainty.

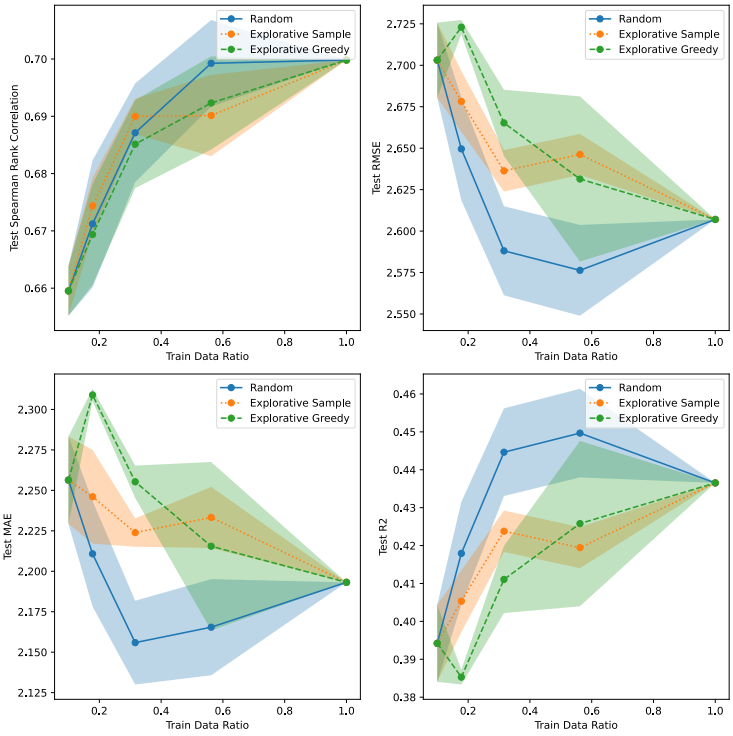

Figure S6: Active learning results for AAV/Sampled vs. Designed using CNN Ensemble uncertainty.

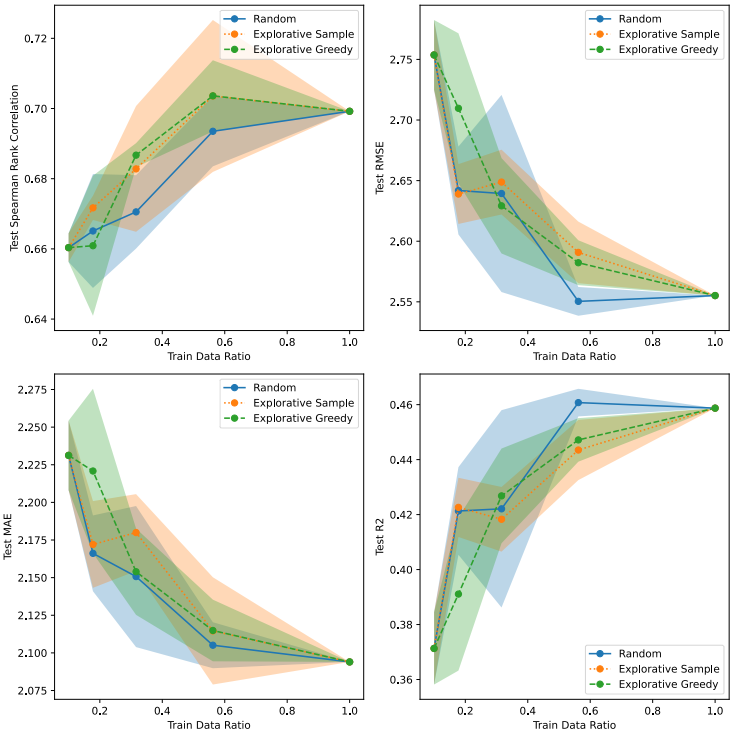

Figure S7: Active learning results for AAV/Sampled vs. Designed using CNN Evidential uncertainty.

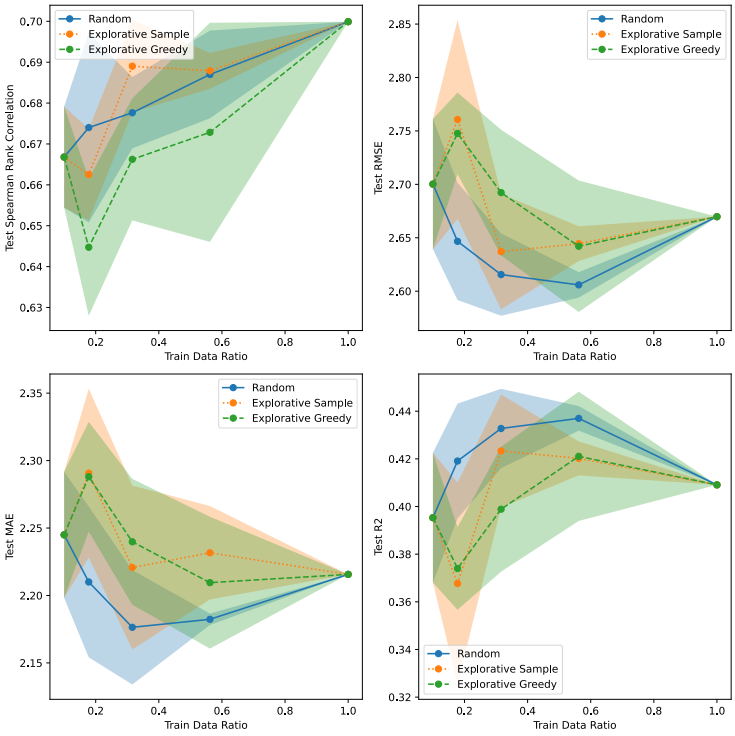

Figure S8: Active learning results for AAV/Sampled vs. Designed using CNN MVE uncertainty.

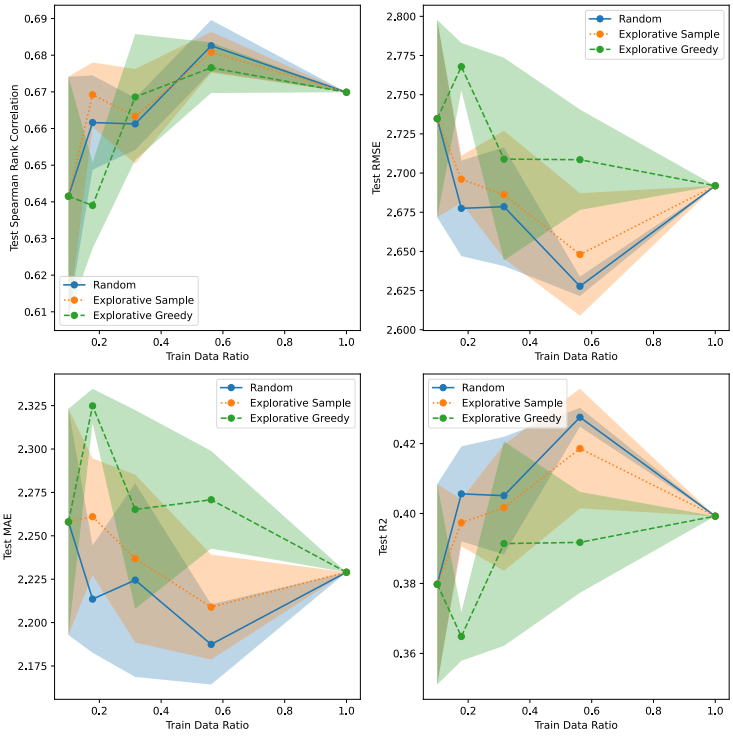

Figure S9: Active learning results for AAV/Sampled vs. Designed using CNN SVI uncertainty.

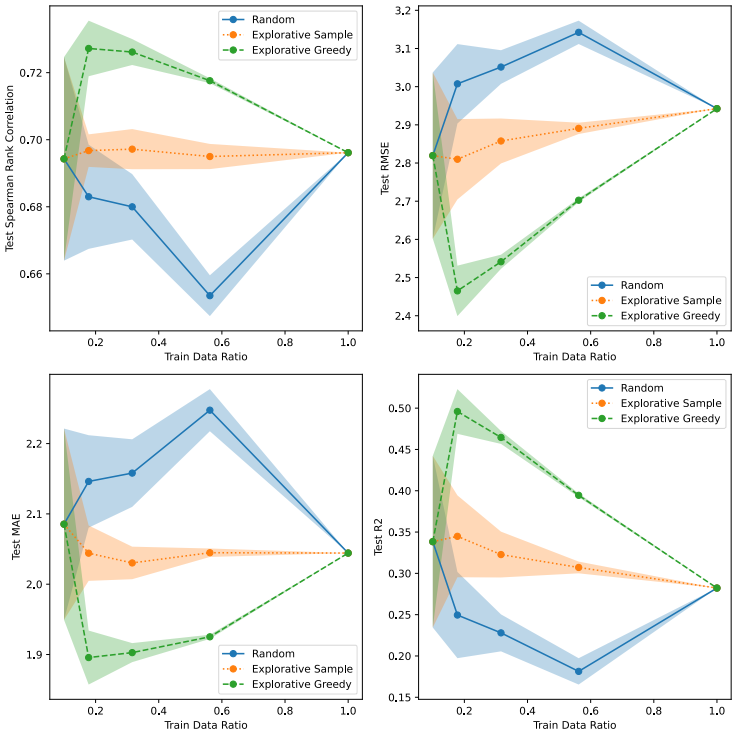

Figure S10: Active learning results for AAV/Sampled vs. Designed using Linear Bayesian Ridge uncertainty.

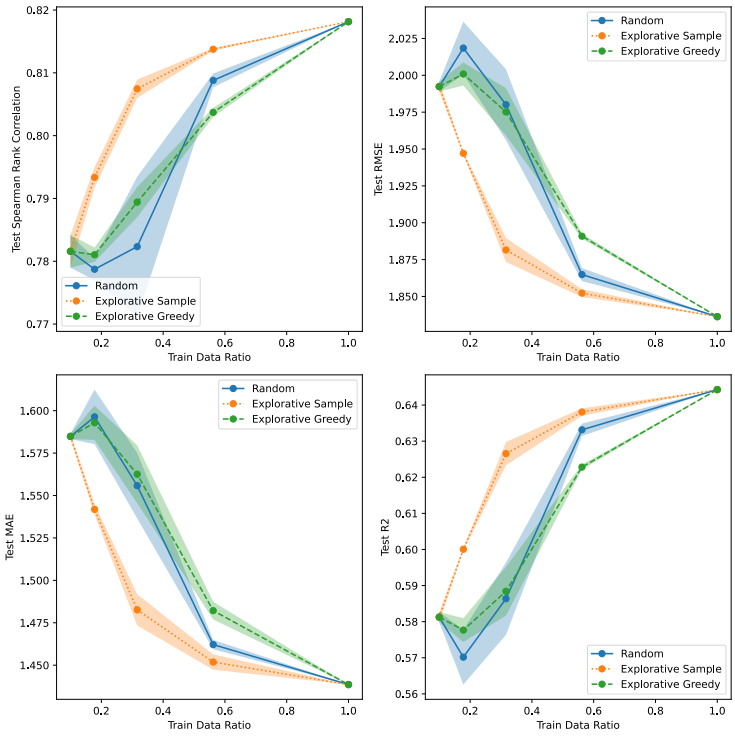

Figure S11: Active learning results for AAV/Random using CNN Dropout uncertainty.

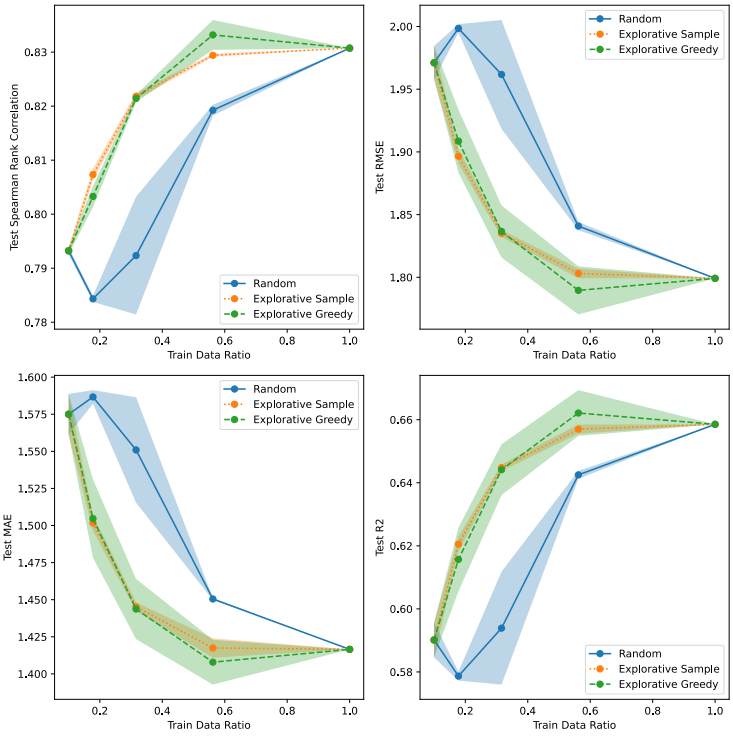

Figure S12: Active learning results for AAV/Random using CNN Ensemble uncertainty.

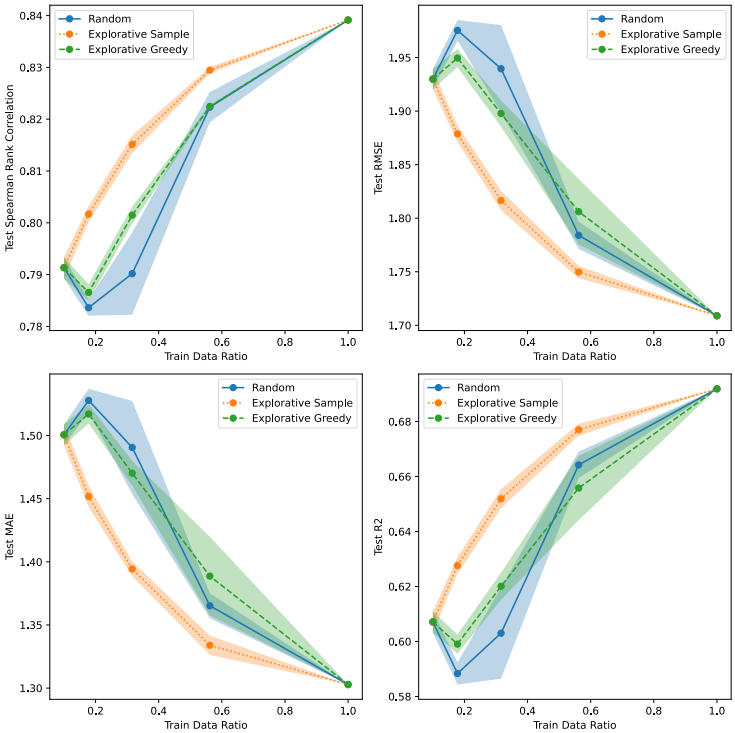

Figure S13: Active learning results for AAV/Random using CNN Evidential uncertainty.

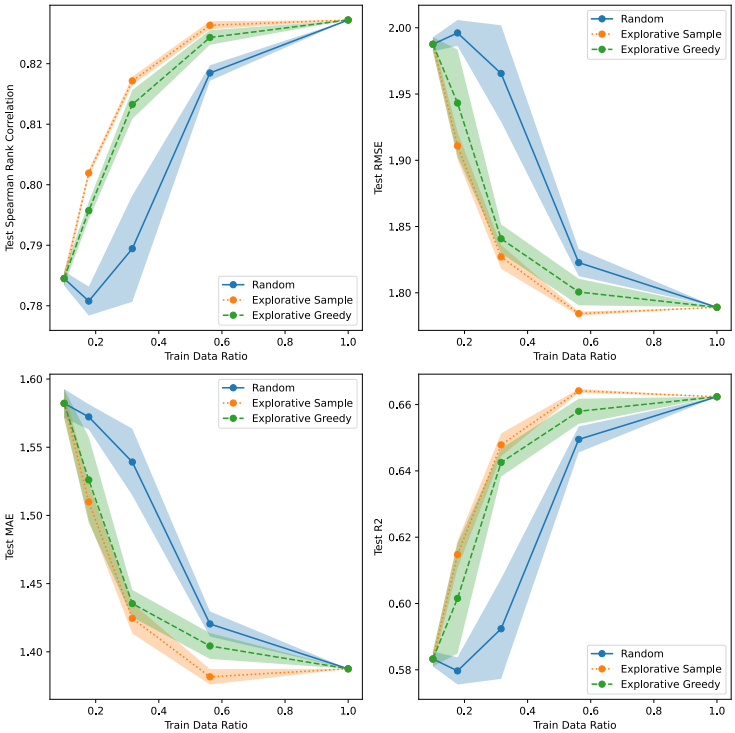

Figure S14: Active learning results for AAV/Random using CNN MVE uncertainty.

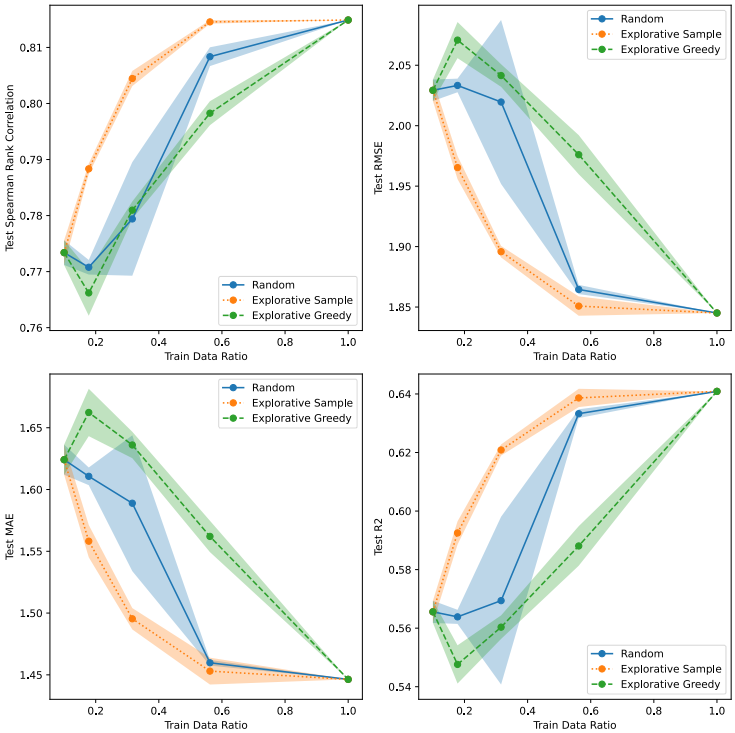

Figure S15: Active learning results for AAV/Random using CNN SVI uncertainty.

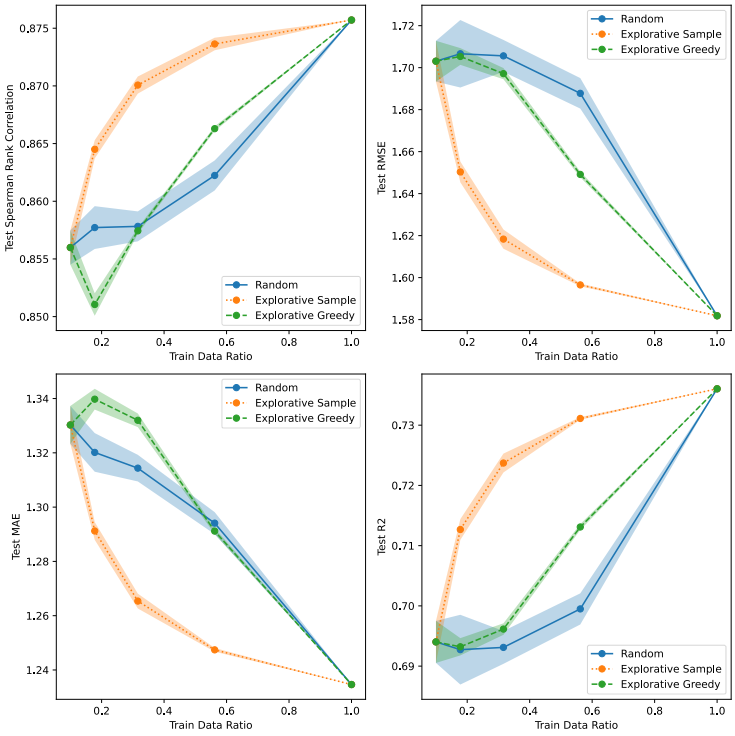

Figure S16: Active learning results for AAV/Random using Linear Bayesian Ridge uncertainty.

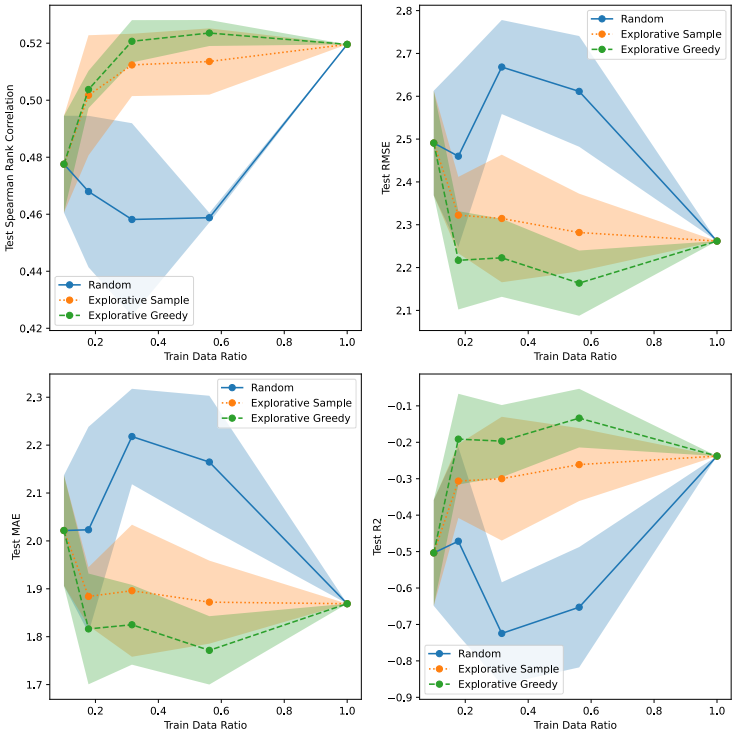

Figure S17: Active learning results for AAV/7 vs. Rest using CNN Dropout uncertainty.

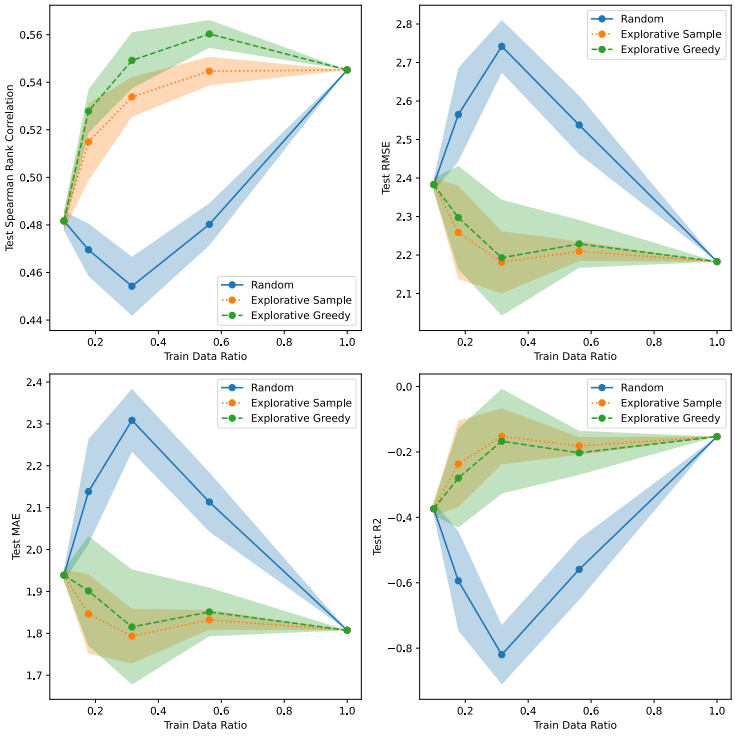

Figure S18: Active learning results for AAV/7 vs. Rest using CNN Ensemble uncertainty.

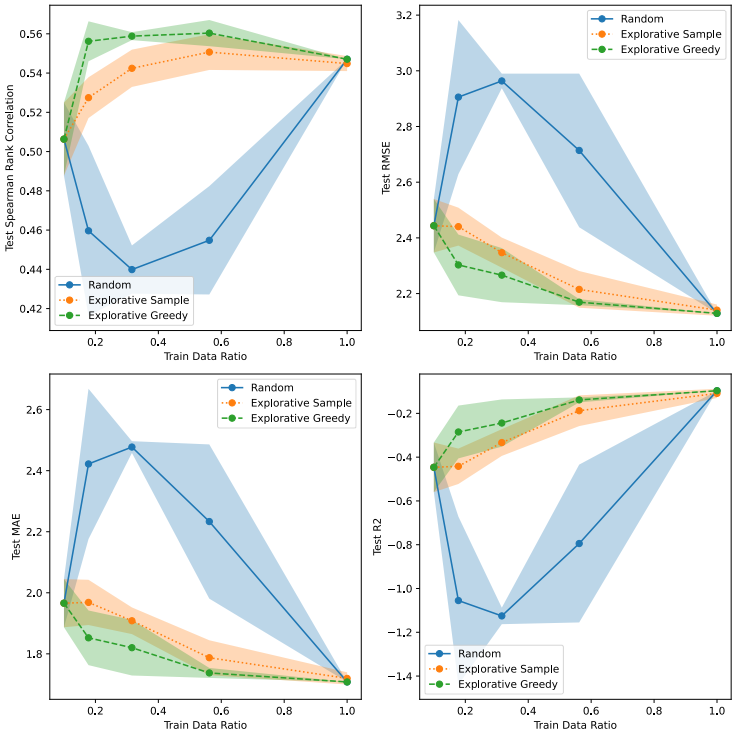

Figure S19: Active learning results for AAV/7 vs. Rest using CNN Evidential uncertainty.

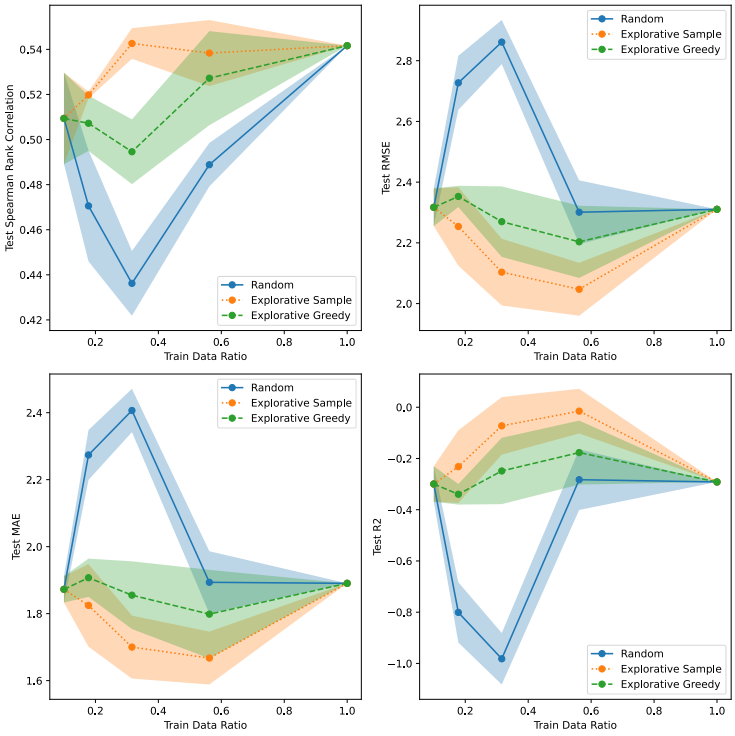

Figure S20: Active learning results for AAV/7 vs. Rest using CNN MVE uncertainty.

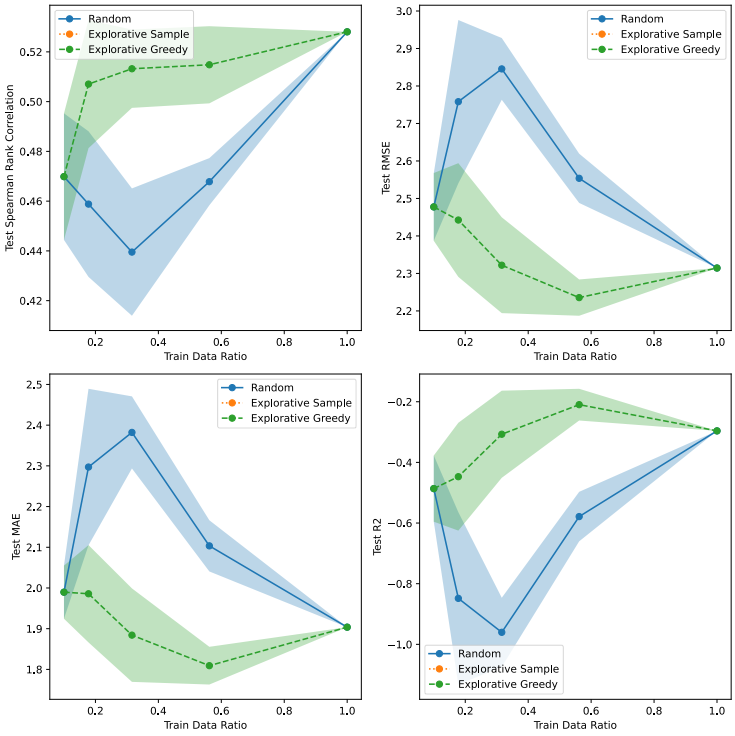

Figure S21: Active learning results for AAV/7 vs. Rest using CNN SVI uncertainty.

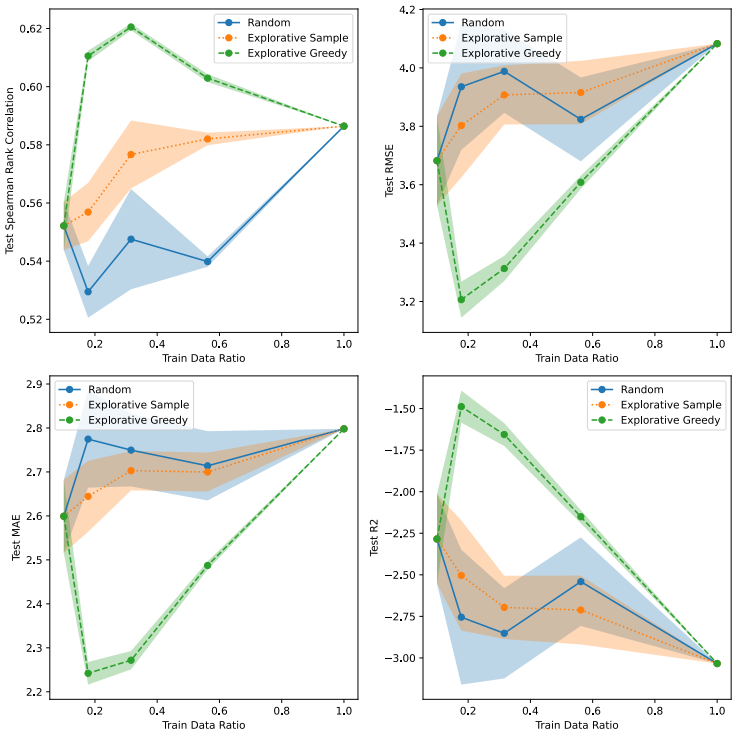

Figure S22: Active learning results for AAV/7 vs. Rest using Linear Bayesian Ridge uncertainty.

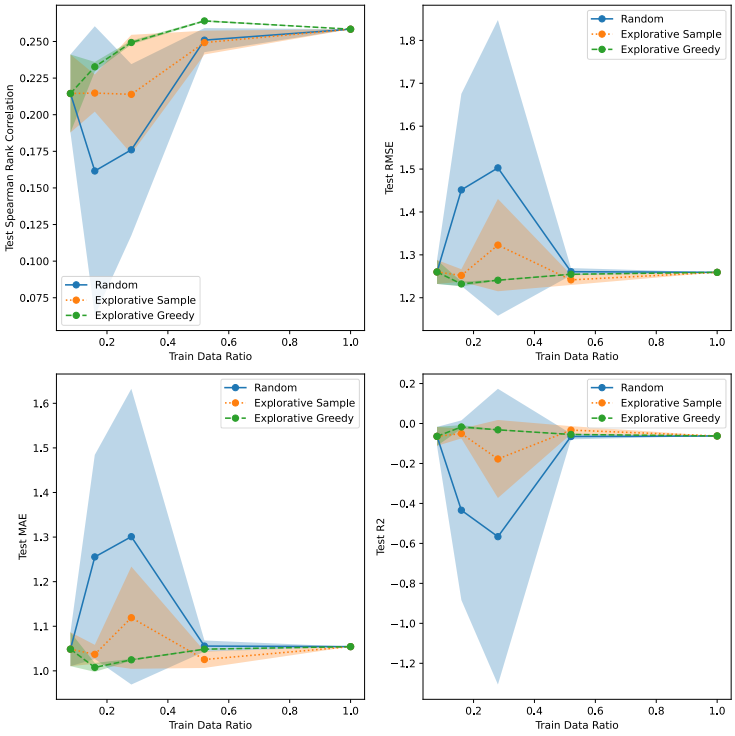

Figure S23: Active learning results for GB1/1 vs. Rest using CNN Dropout uncertainty.

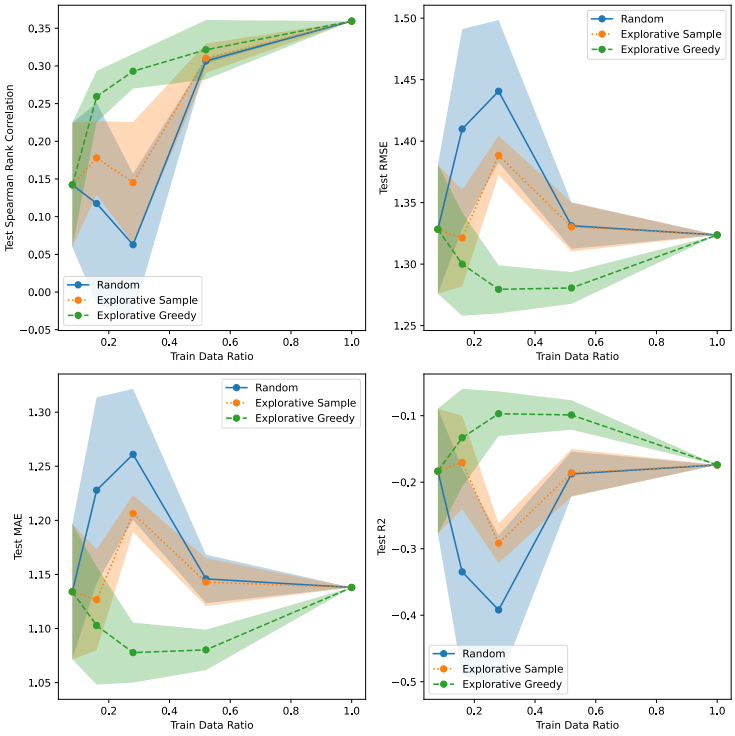

Figure S24: Active learning results for GB1/1 vs. Rest using CNN Ensemble uncertainty.

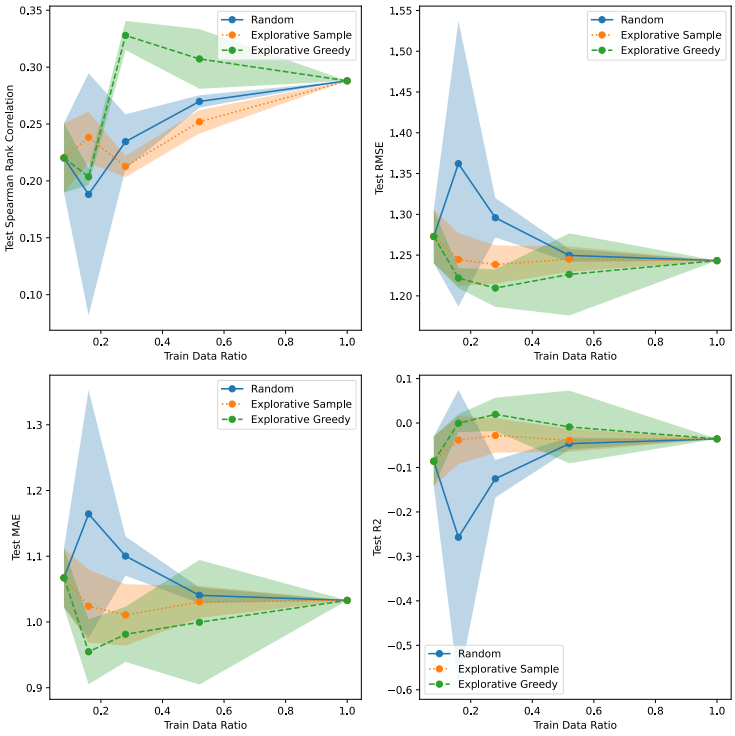

Figure S25: Active learning results for GB1/1 vs. Rest using CNN Evidential uncertainty.

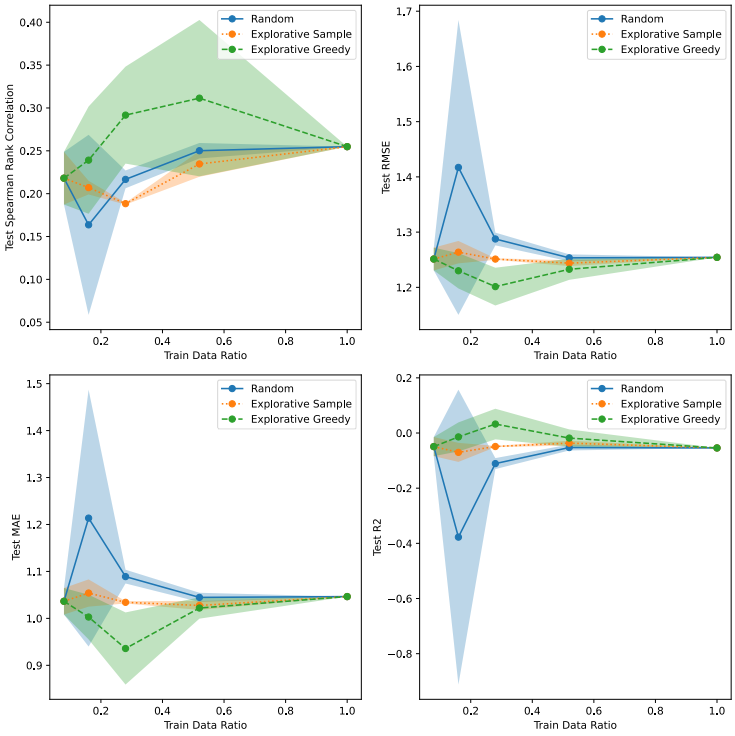

Figure S26: Active learning results for GB1/1 vs. Rest using CNN MVE uncertainty.

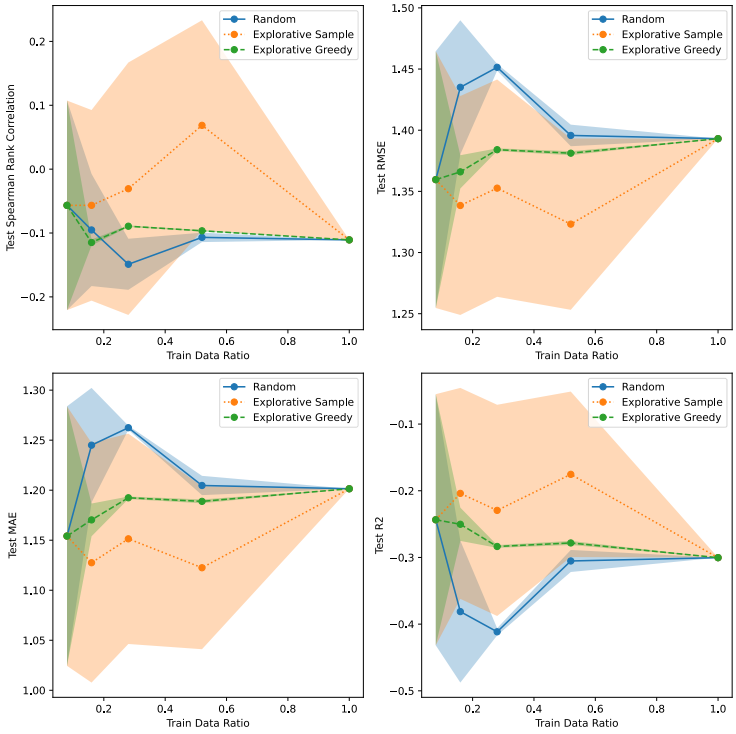

Figure S27: Active learning results for GB1/1 vs. Rest using CNN SVI uncertainty.

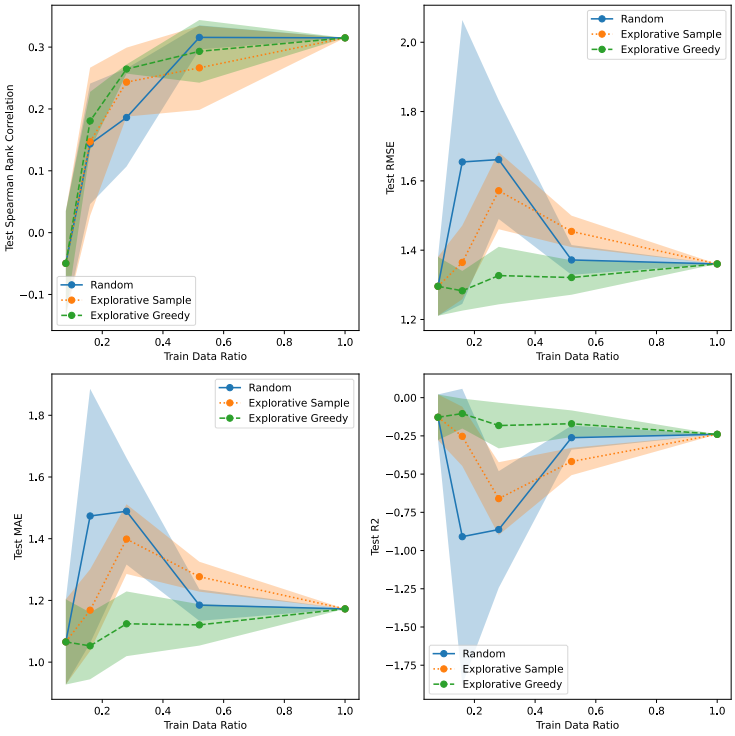

Figure S28: Active learning results for GB1/1 vs. Rest using GP Continuous uncertainty.

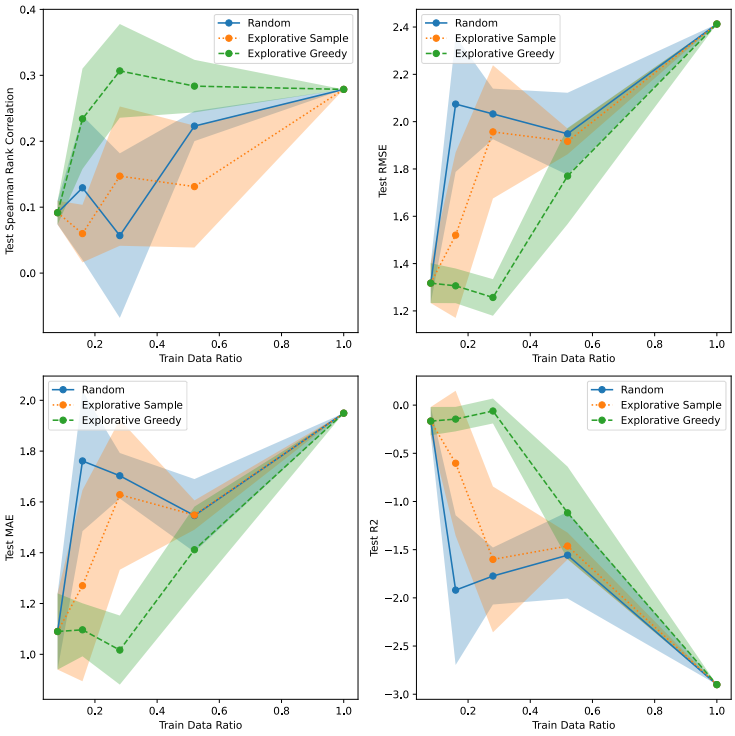

Figure S29: Active learning results for GB1/1 vs. Rest using Linear Bayesian Ridge uncertainty.

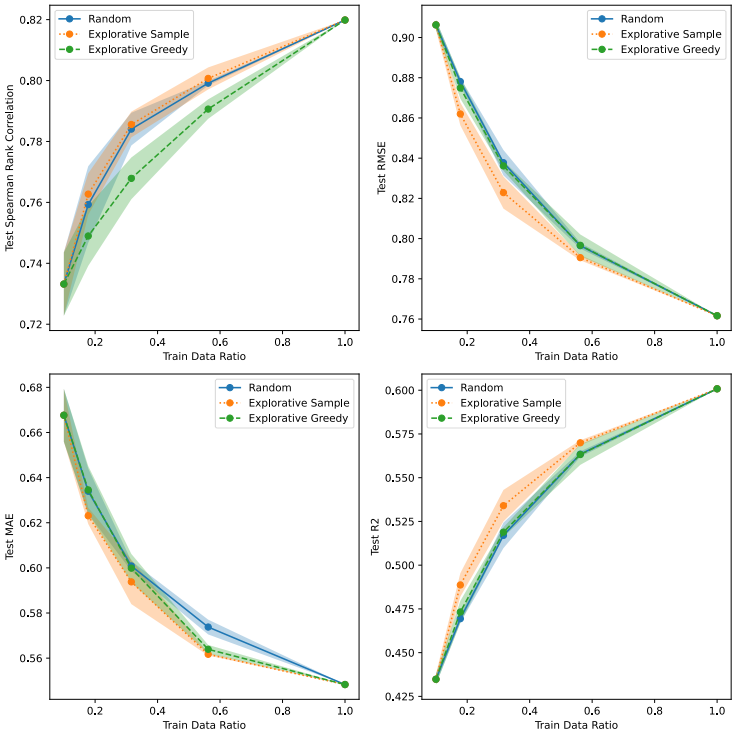

Figure S30: Active learning results for GB1/Random using CNN Dropout uncertainty.

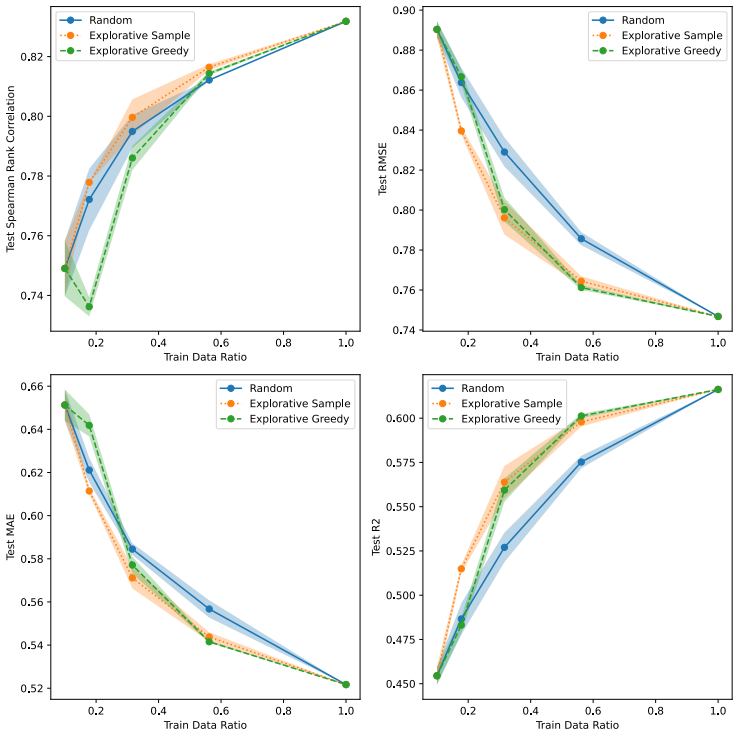

Figure S31: Active learning results for GB1/Random using CNN Ensemble uncertainty.

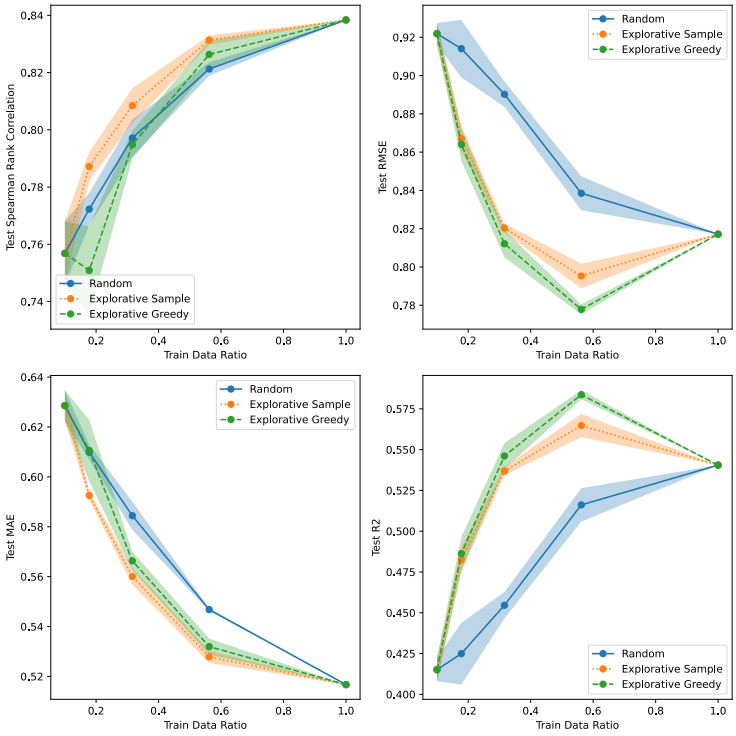

Figure S32: Active learning results for GB1/Random using CNN Evidential uncertainty.

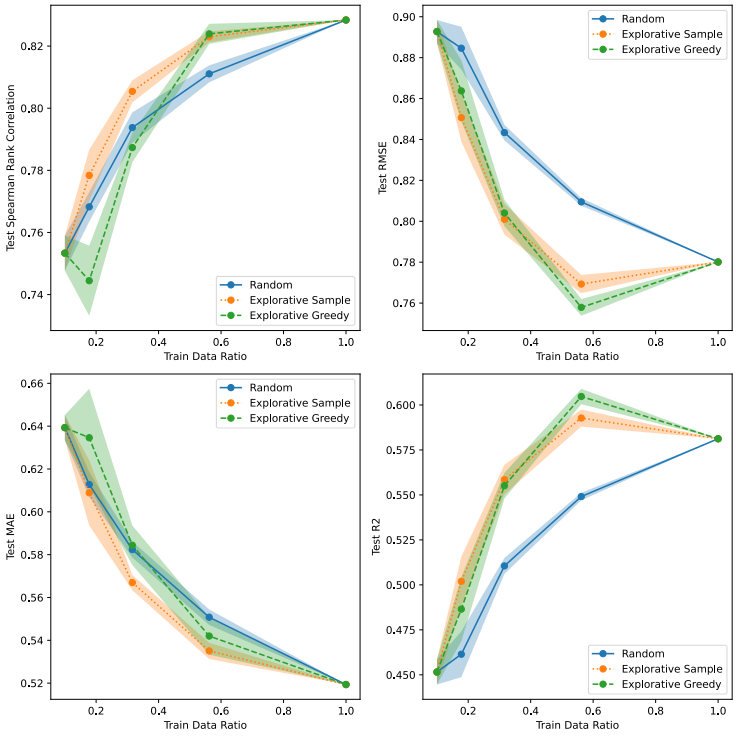

Figure S33: Active learning results for GB1/Random using CNN MVE uncertainty.

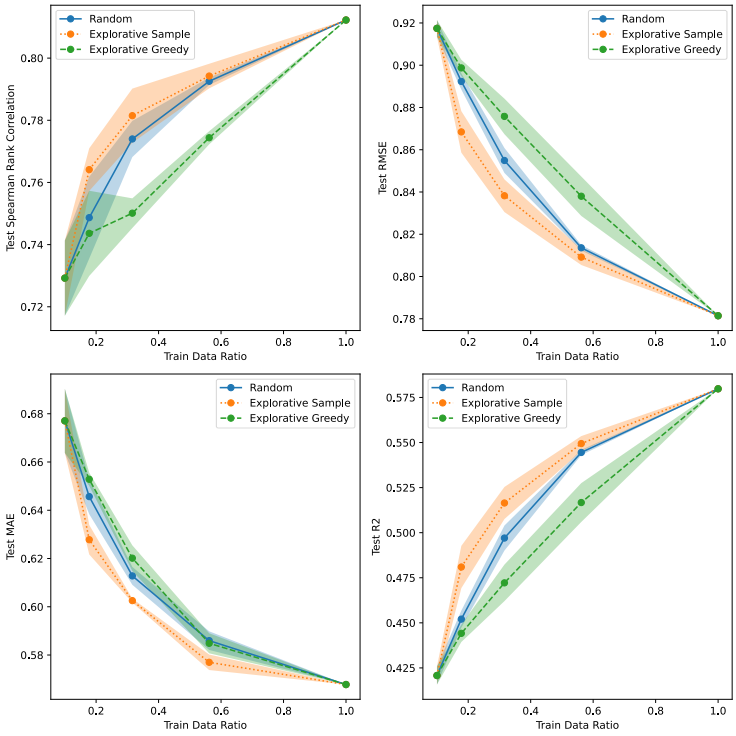

Figure S34: Active learning results for GB1/Random using CNN SVI uncertainty.

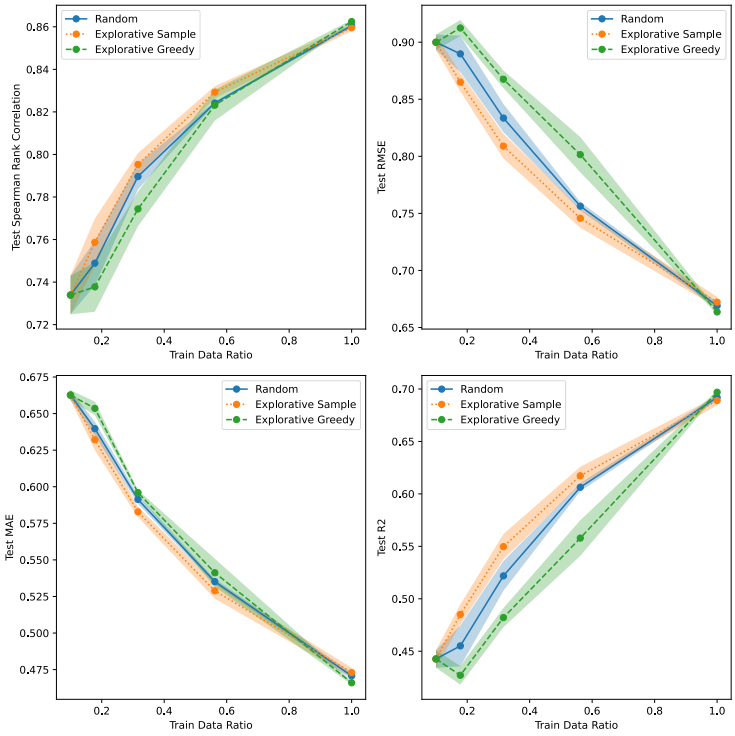

Figure S35: Active learning results for GB1/Random using GP Continuous uncertainty.

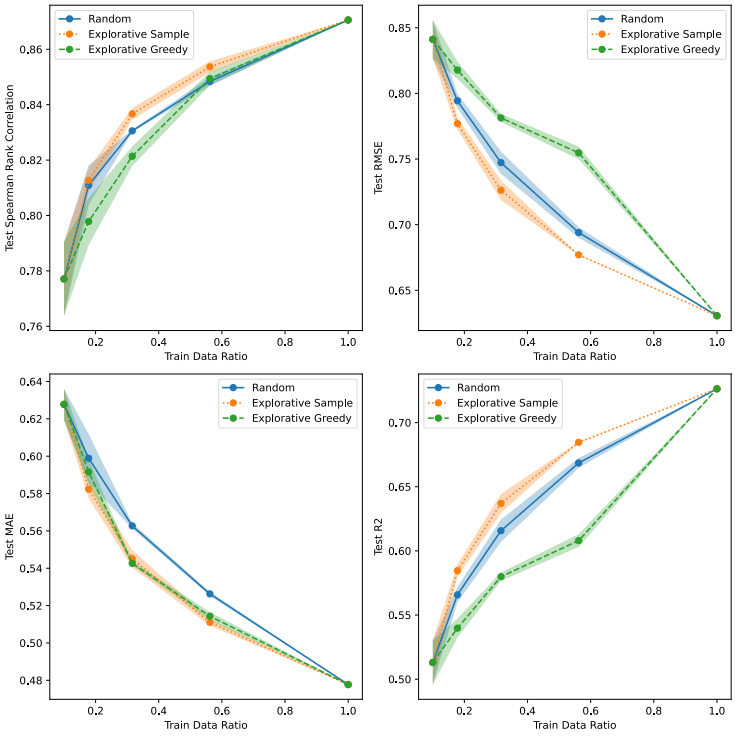

Figure S36: Active learning results for GB1/Random using Linear Bayesian Ridge uncertainty.

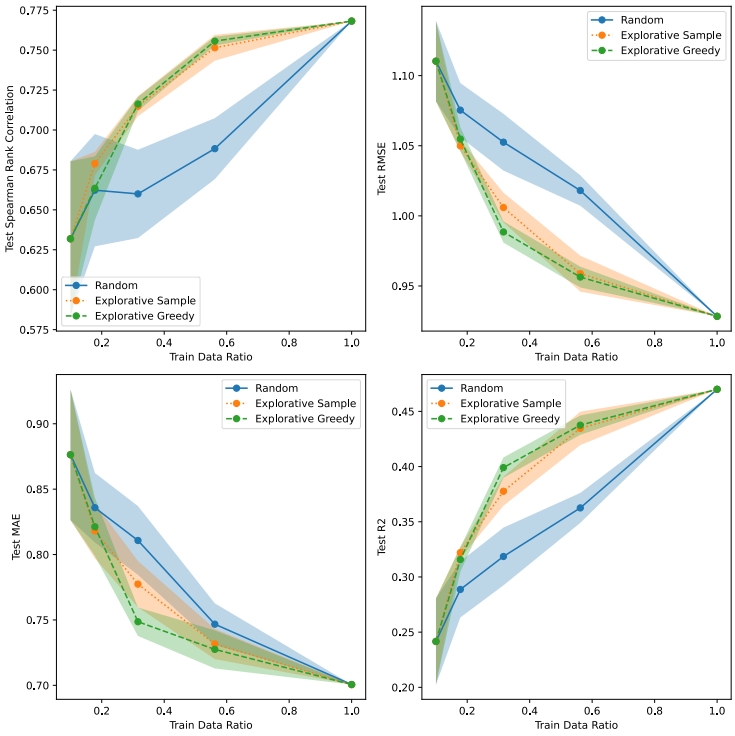

Figure S37: Active learning results for GB1/3 vs. Rest using CNN Dropout uncertainty.

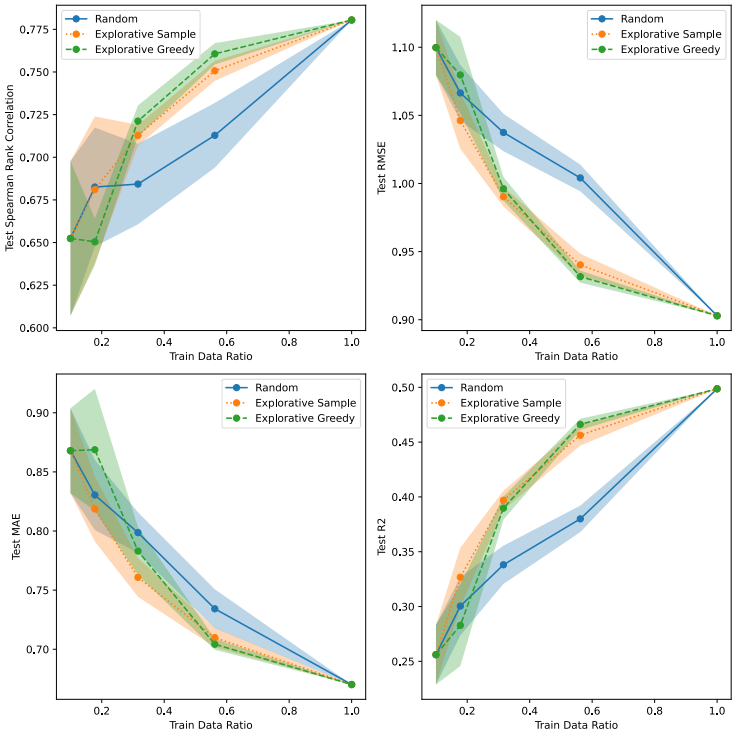

Figure S38: Active learning results for GB1/3 vs. Rest using CNN Ensemble uncertainty.

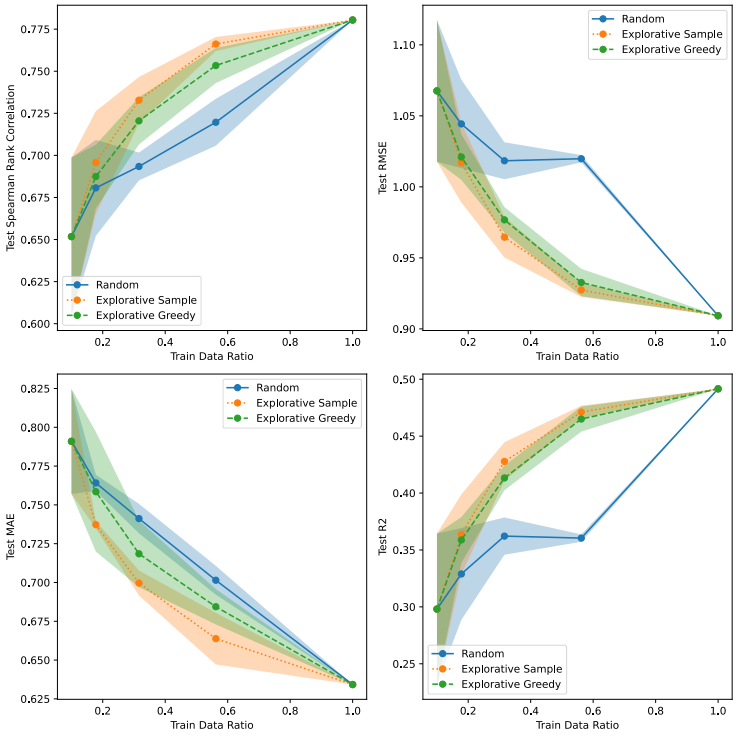

Figure S39: Active learning results for GB1/3 vs. Rest using CNN Evidential uncertainty.

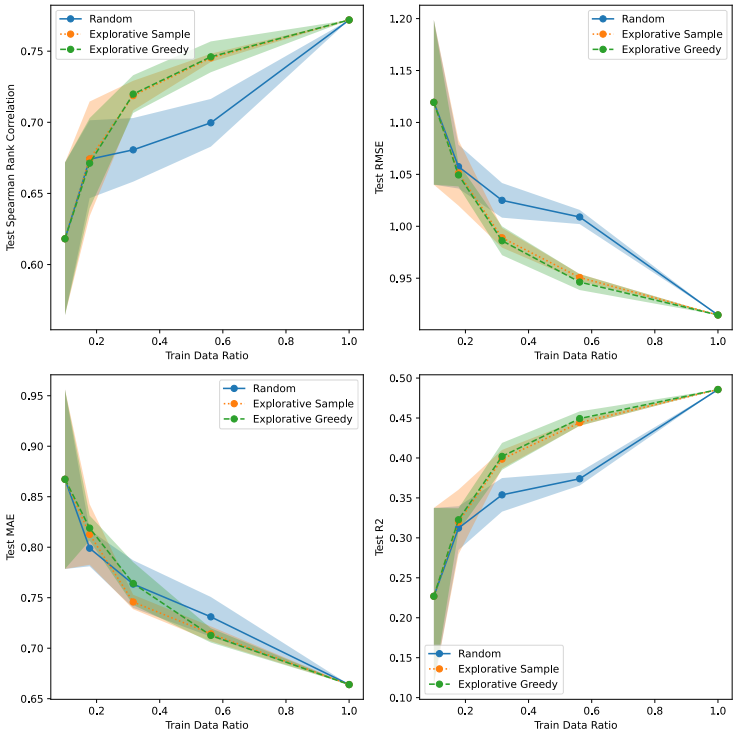

Figure S40: Active learning results for GB1/3 vs. Rest using CNN MVE uncertainty.

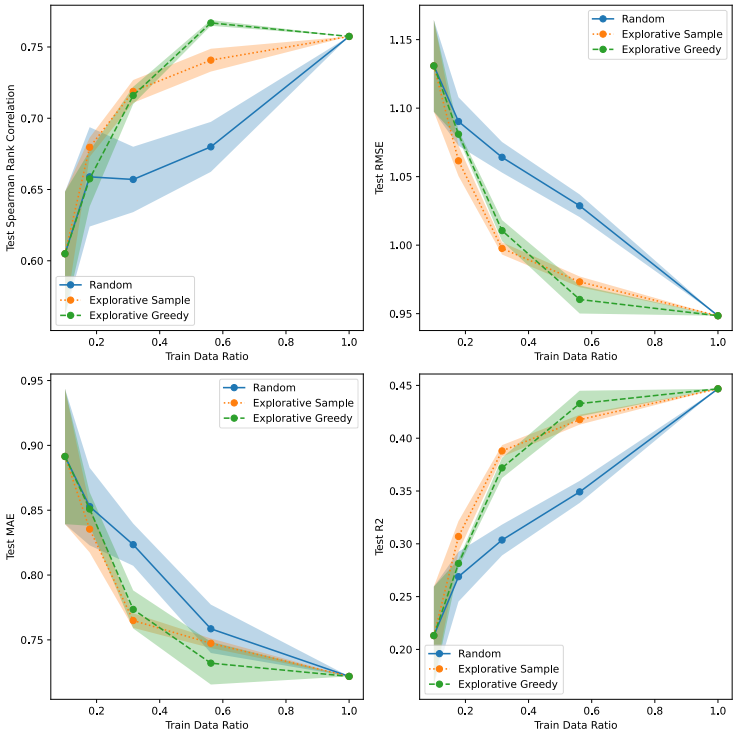

Figure S41: Active learning results for GB1/3 vs. Rest using CNN SVI uncertainty.

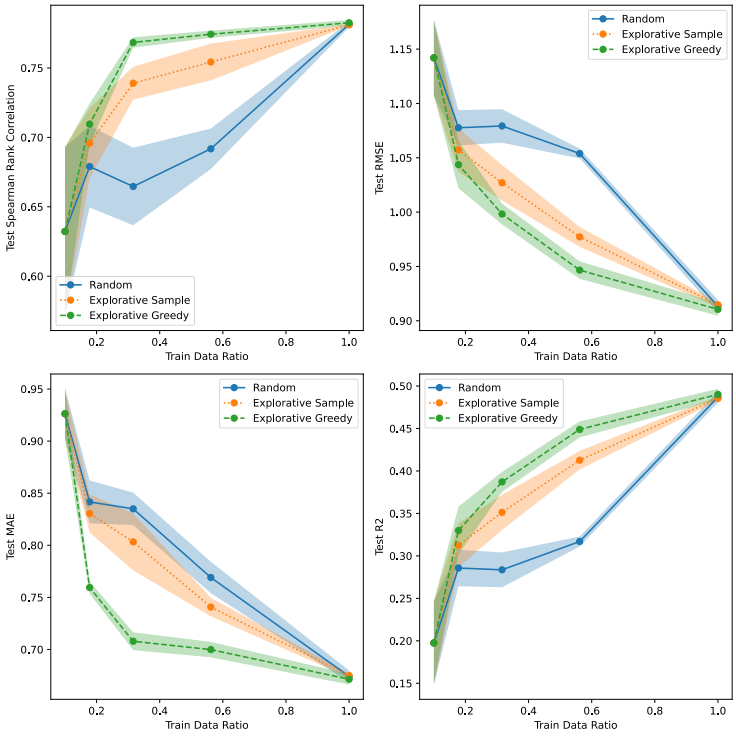

Figure S42: Active learning results for GB1/3 vs. Rest using GP Continuous uncertainty.

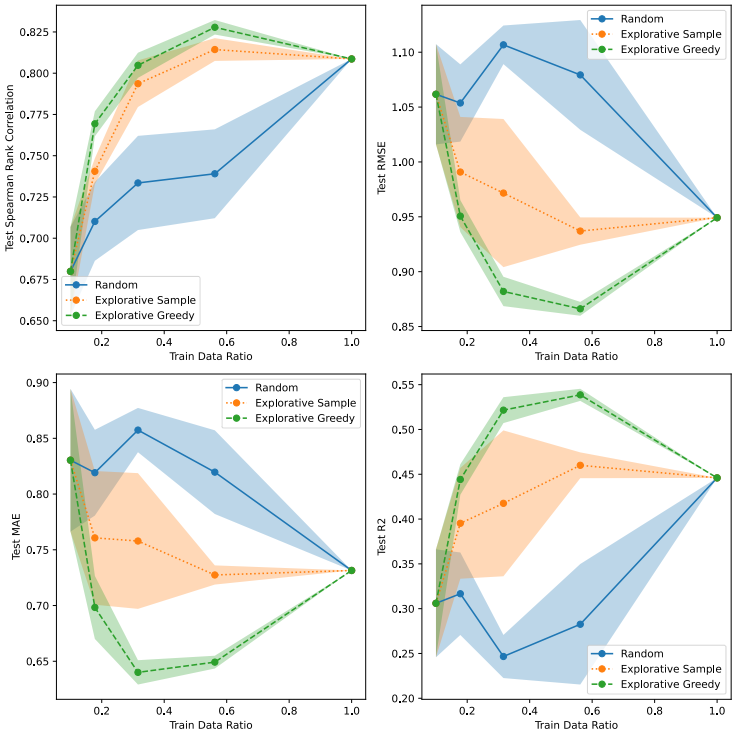

Figure S43: Active learning results for GB1/3 vs. Rest using Linear Bayesian Ridge uncertainty.

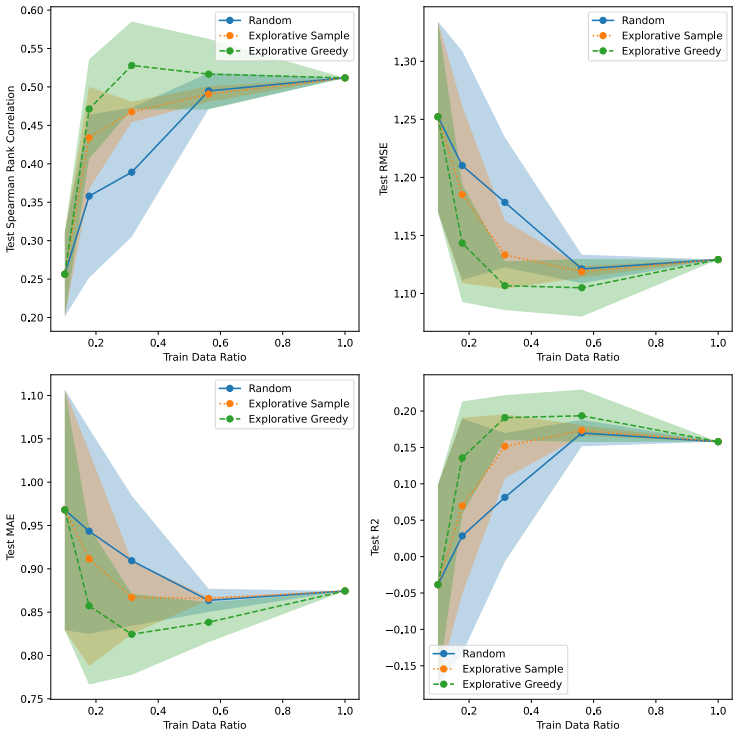

Figure S44: Active learning results for GB1/2 vs. Rest using CNN Dropout uncertainty.

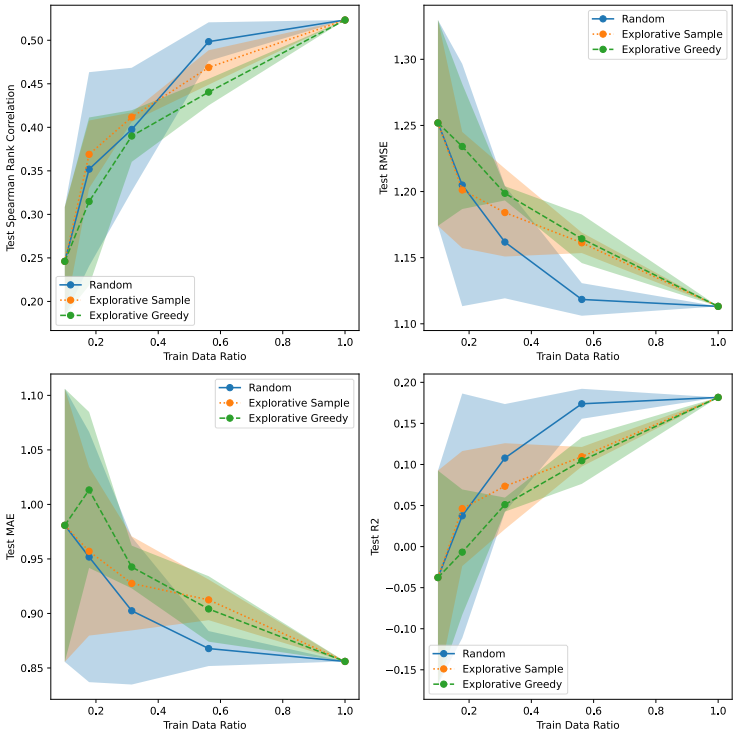

Figure S45: Active learning results for GB1/2 vs. Rest using CNN Ensemble uncertainty.

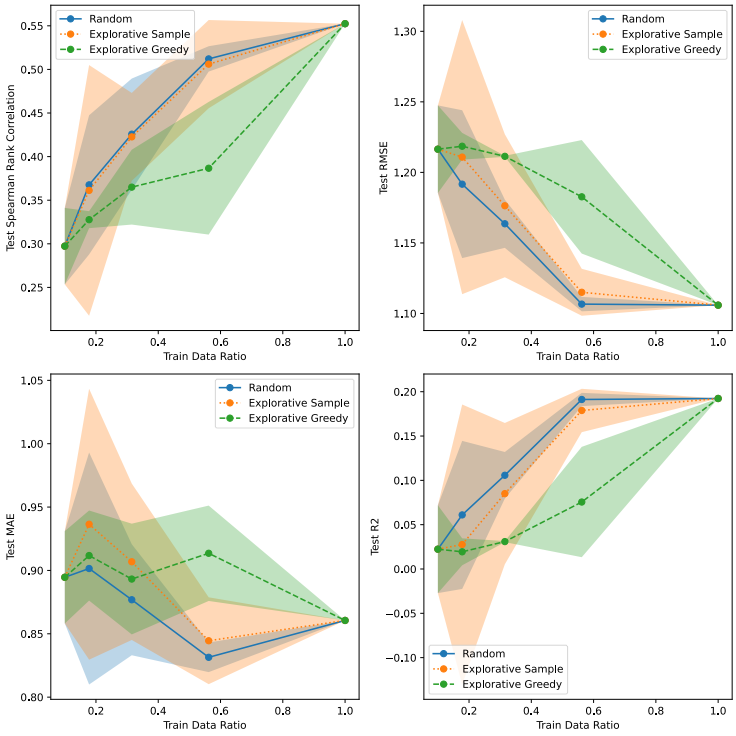

Figure S46: Active learning results for GB1/2 vs. Rest using CNN Evidential uncertainty.

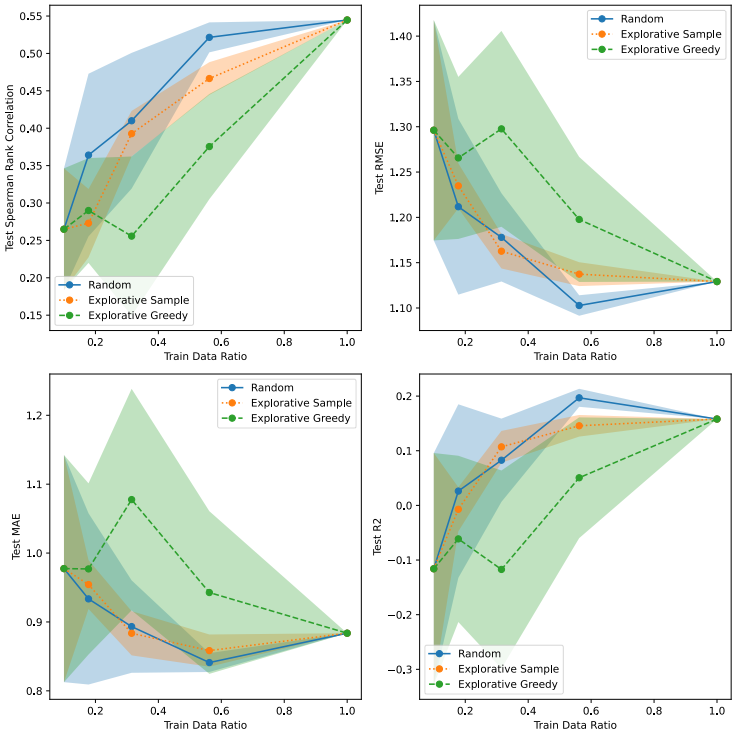

Figure S47: Active learning results for GB1/2 vs. Rest using CNN MVE uncertainty.

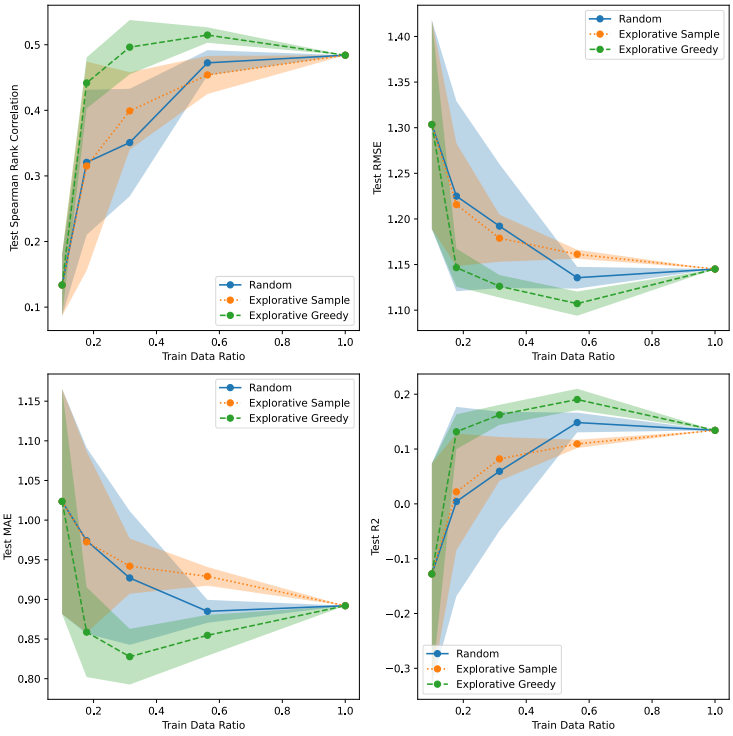

Figure S48: Active learning results for GB1/2 vs. Rest using CNN SVI uncertainty.

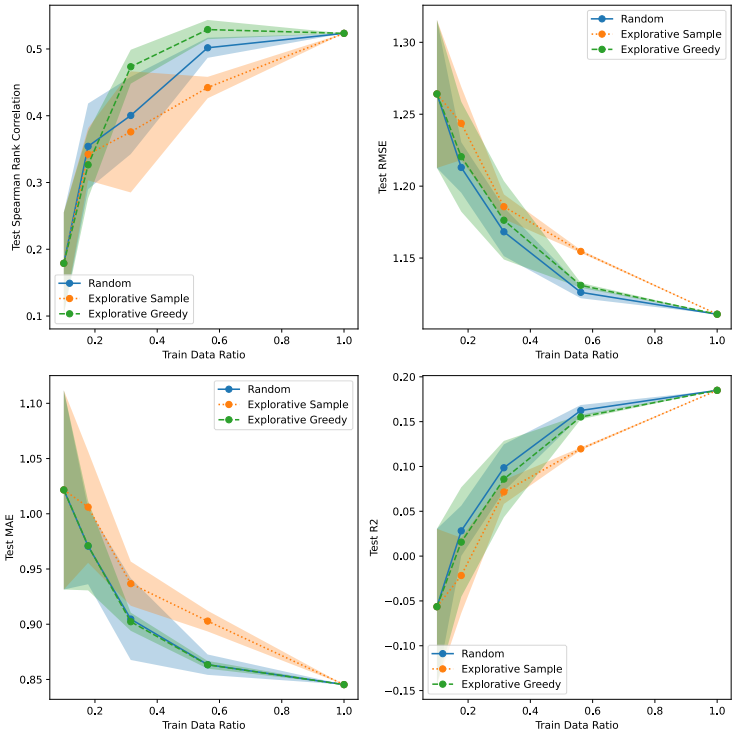

Figure S49: Active learning results for GB1/2 vs. Rest using GP Continuous uncertainty.

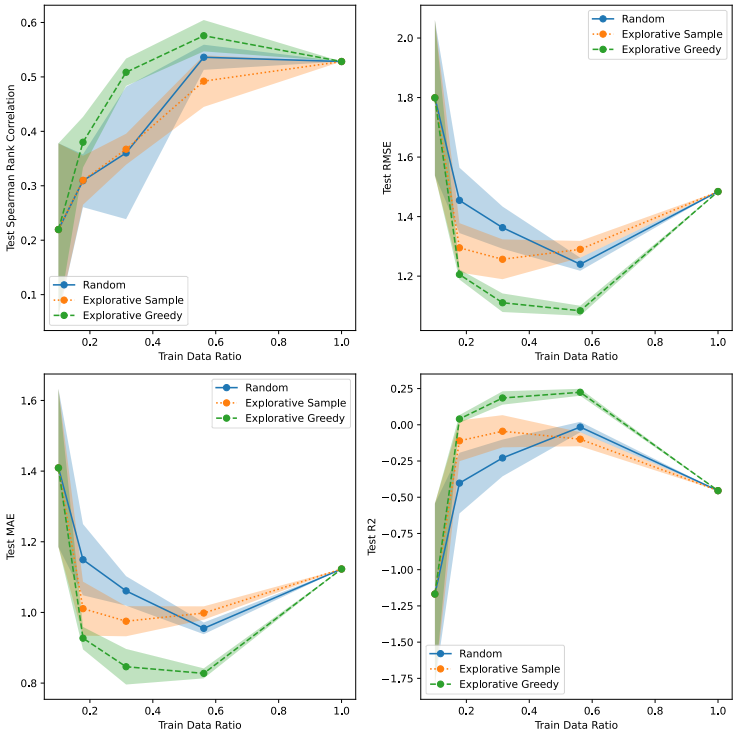

Figure S50: Active learning results for GB1/2 vs. Rest using Linear Bayesian Ridge uncertainty.

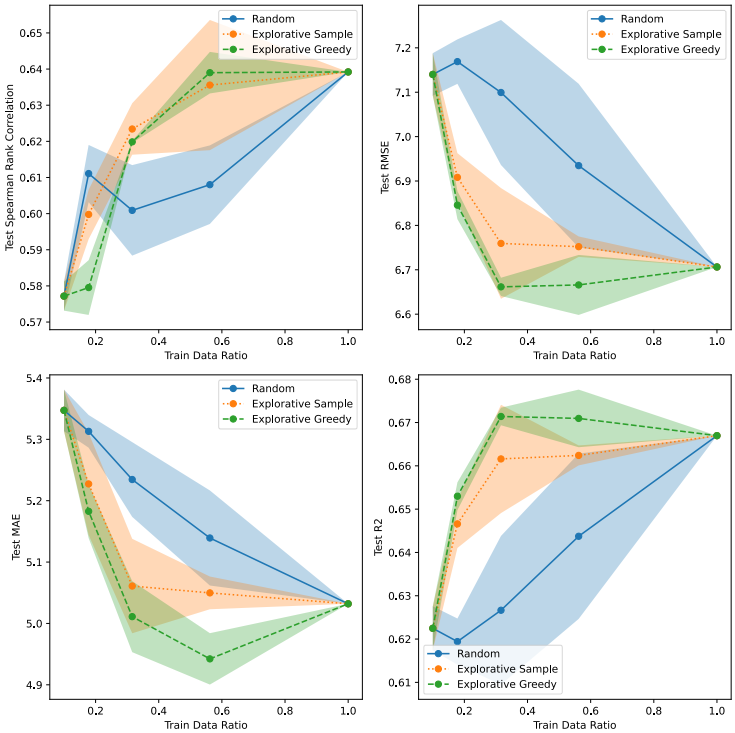

Figure S51: Active learning results for Meltome/Random using CNN Dropout uncertainty.

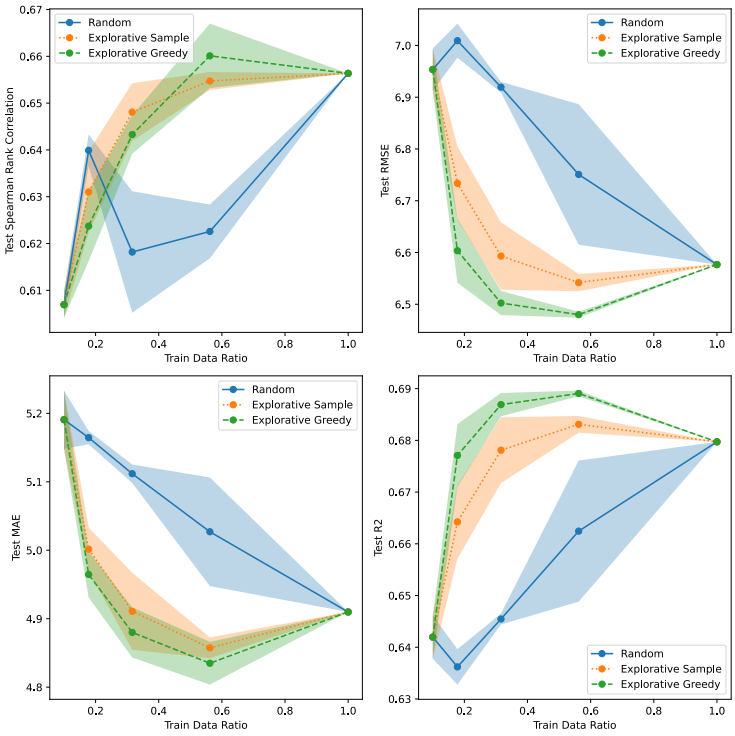

Figure S52: Active learning results for Meltome/Random using CNN Ensemble uncertainty.

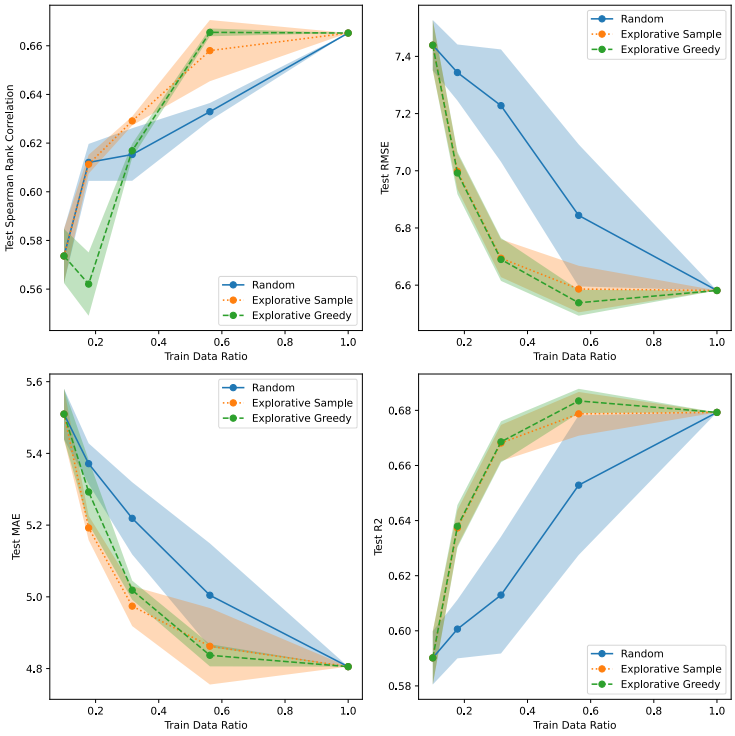

Figure S53: Active learning results for Meltome/Random using CNN Evidential uncertainty.

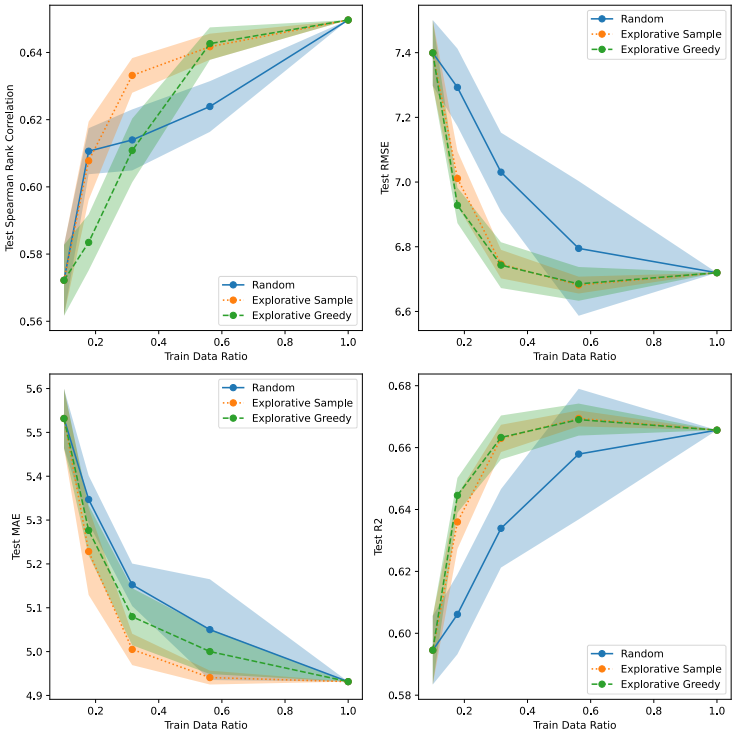

Figure S54: Active learning results for Meltome/Random using CNN MVE uncertainty.

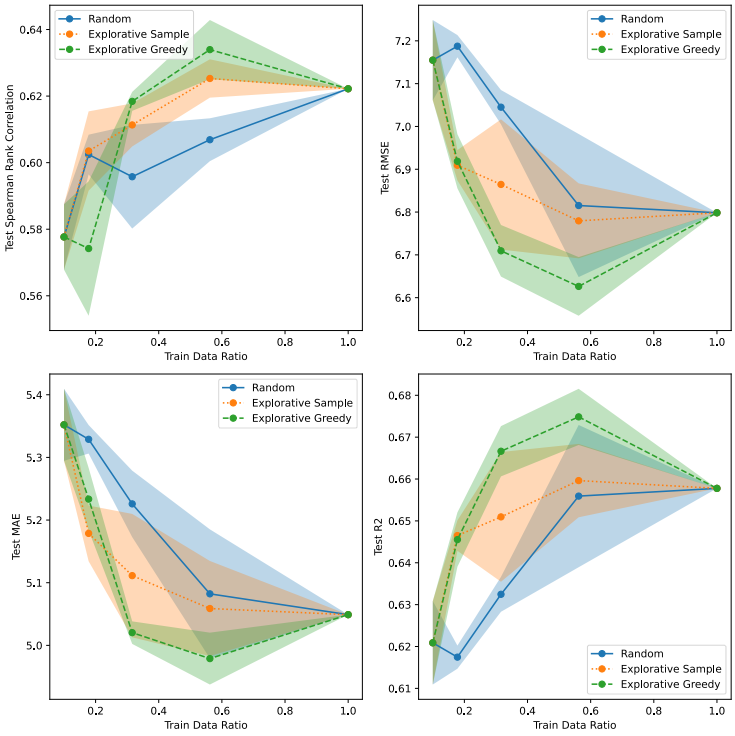

Figure S55: Active learning results for Meltome/Random using CNN SVI uncertainty.

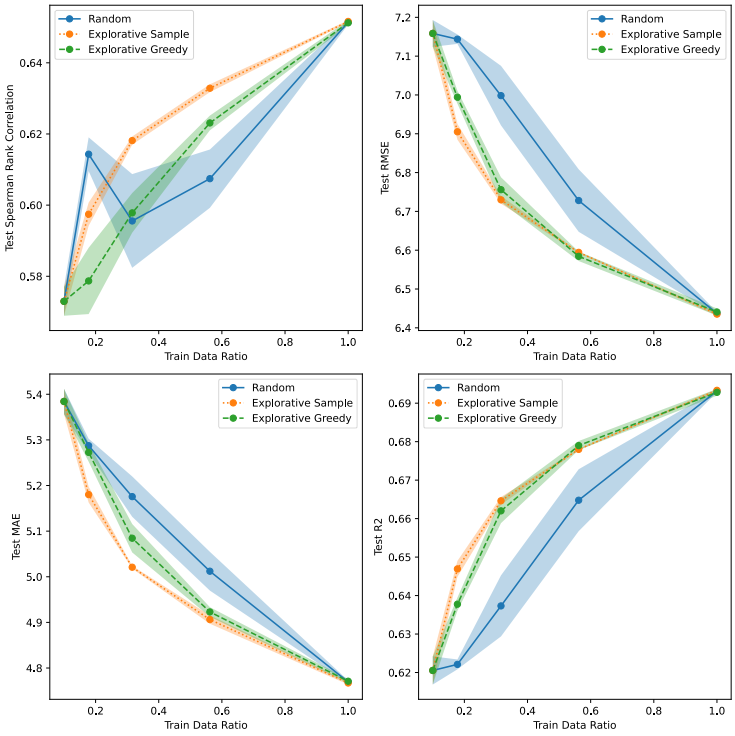

Figure S56: Active learning results for Meltome/Random using GP Continuous uncertainty.

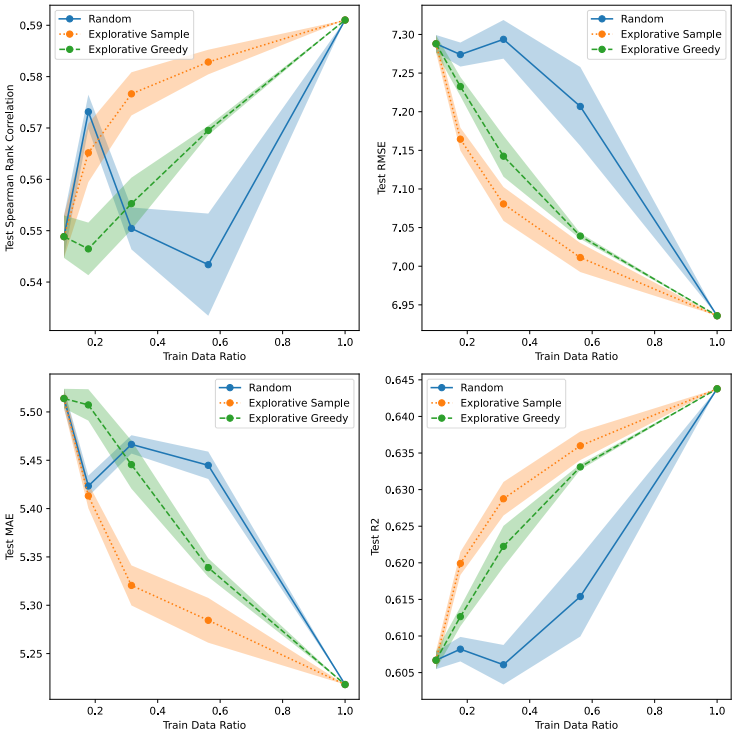

Figure S57: Active learning results for Meltome/Random using Linear Bayesian Ridge uncertainty.