On minimizing the training set fill distance in machine learning regression

Paolo Climaco[†] CLIMACO@INS.UNI-BONN.DE
Jochen Garcke [†,‡] GARCKE@INS.UNI-BONN.DE
[†]*Institut für Numerische Simulation, Universität Bonn, Germany*
[‡]*Fraunhofer SCAI, Sankt Augustin, Germany*

Reviewed on OpenReview: *https://openreview.net/forum?id=XXXX*

Editor: My editor

Abstract

For regression tasks one often leverages large datasets for training predictive machine learning models. However, using large datasets may not be feasible due to computational limitations or high data labelling costs. Therefore, suitably selecting small training sets from large pools of unlabelled data points is essential to maximize model performance while maintaining efficiency. In this work, we study Farthest Point Sampling (FPS), a data selection approach that aims to minimize the fill distance of the selected set. We derive an upper bound for the maximum expected prediction error, conditional to the location of the unlabelled data points, that linearly depends on the training set fill distance. For empirical validation, we perform experiments using two regression models on three datasets. We empirically show that selecting a training set by aiming to minimize the fill distance, thereby minimizing our derived bound, significantly reduces the maximum prediction error of various regression models, outperforming alternative sampling approaches by a large margin. Furthermore, we show that selecting training sets with the FPS can also increase model stability for the specific case of Gaussian kernel regression approaches.

Keywords: Fill distance, Farthest Point Sampling, Regression.

1 Introduction

Machine learning (ML) regression models are widely used in applications, where we are in particular interested in molecular property prediction (Montavon et al., 2013; Hansen et al., 2015). One of the main goals of ML regression is to label, with continuous values, pools of unlabelled data points for which the existing labelling methods, e.g., numerical simulations or laboratory experiments, are too expensive in terms of computation, time, or money. To achieve this, a subset of the unlabelled pool is labelled and used to train a ML model, which is then employed to get fast predictions for the labels of points not considered during training. However, the effectiveness of ML regression models is strongly dependent on the training data used for learning. Therefore, the selection of a suitable training set is crucial for the quality of the predictions of the model. Our focus is on selecting data points that result in a good performance for a variety of regression models. This ansatz ensures that the labelling effort is not wasted on subsets that may only be useful for specific learning models, classes of models, or prediction tasks.

We distinguish between active and passive dataset selection strategies. Active learning (Settles, 2012) involves learning one or several regression models, predicting uncertainties for unlabelled data, based on which the most uncertain ones are selected for labelling and the cycle starts anew, until (qualitative) stopping criteria are fulfilled. Unfortunately, it typically only benefits a specific model or model class and optimizes the performance of the models for a specific learning task, as it exploits the knowledge of the labels to iteratively update the parameters of the models during the selection process. Passive sampling (Yu and Kim, 2010) is based only on the feature space locations. Consequently, it has the potential to offer advantages when considering multiple learning tasks that pertain to the same data, as it is independent of the label values associated with the analyzed data points. We think passive sampling can be further divided into two subclasses: model-dependent and model-agnostic. Model dependent passive sampling strategies are developed to benefit specific learning models or model classes, such as linear regression (Yu et al., 2006), k-nearest neighbors, or naive Bayes (Wei et al., 2015), similar to active learning. Contrarily, model-agnostic strategies have the potential to benefit multiple classes of regression models rather than just one. Farthest point sampling (FPS) (Eldar et al., 1994) is a well-established passive sampling model-agnostic strategy for training set selection already employed in various application fields, such as image classification (Sener and Savarese, 2018) or chemical and material science (Deringer et al., 2021). FPS provides suboptimal solutions to the k-center problem (Har-Peled, 2011), which involves selecting a subset of k points from a given set by minimizing the fill distance of the selected set, that is, the maximum over the distances between any point in the remaining set and selected point nearest to it.

Our study aims to investigate theoretically and empirically the impact of minimizing training set fill distance through FPS for ML regression. For classification tasks, it was shown that minimizing the fill distance of the training set reduces the average prediction error of Lipschitz-continuous classification models with soft-max output layer and bounded error function (Sener and Savarese, 2018). Unfortunately, these results do not carry over to regression tasks, even for simpler Lipschitz-continuous approaches, such as kernel ridge regression with the Gaussian kernel (KRR) or feed-forward neural networks (FNNs). In particular, we provide examples where reducing the training set fill distance does not significantly lower the average prediction error compared to random selection. The benefits of using FPS in regression have been studied in various works (Yu and Kim, 2010; Wu et al., 2019; Deringer et al., 2021), where it was argued that passive sampling strategies such as FPS are more effective than active learning in terms of data efficiency and prediction accuracy. However, these works lack theoretical motivation, relying only on domain knowledge or heuristics.

In this work, we derive an upper bound for the maximum expected prediction error of Lipschitz continuous regression models that is linearly dependent on the training set fill distance. We show that minimizing the training set fill distance significantly decreases the maximum approximation error of Lipschitz continuous regression models. We compare the FPS approach with other model-agnostic sampling techniques and demonstrate its superiority for low training set budgets in terms of maximum prediction error reduction. The maximum prediction error of a regression model can be considered as a measure of the robustness of the predictions and is a helpful metric in various applications fields, such as material science and chemistry (Zaverkin et al., 2022), where the average error provides an incomplete evaluation of the predictions of a model (Sutton et al., 2020; Gould and Dale, 2022). Our analysis offers theoretical and empirical results, which set it apart from previous works. Specifically, we extend the theoretical work of Sener and Savarese (2018) from classification to regression, demonstrating that reducing training set fill distance lowers the maximum prediction error of the regression model. Moreover, contrary to Yu and Kim (2010) and Wu et al. (2019), who studied the advantages of using FPS for regression tasks, our findings are supported by mathematical results providing theoretical motivation for what we show empirically. We emphasize that, according to our knowledge, prior research did not detect the relationship between reducing the fill distance of the training set using FPS and decreasing the maximum prediction error of a regression model, neither theoretically nor empirically. In addition, we provide further theoretical examination and empirical investigations to show supplementary advantages of selecting training sets with the FPS for kernel regression models using a Gaussian kernel. Specifically, our findings indicate that employing FPS for selecting training sets enhances the stability of this particular category of models.

2 Related work

Existing work concerning model-agnostic passive sampling is mostly related to coresets approaches. Coresets (Feldman, 2019) aim to identify the most informative training data subset, according to some principle. The simplest coreset method is uniform sampling, which randomly selects subsets from the given pool of data points. Importance sampling approaches, such as the CUR algorithm (Mahoney and Drineas, 2009), assign to samples relevance-based weights. Cluster-besed methods such as k-medoids and k-medoids++ (Mannor et al., 2011), that are adapted version of k-means and k-means++ (Murphy, 2022), segment the feature space in clusters and select representative points from each cluster. Greedy algorithms iteratively select the most informative data points based on a predefined criterion. Well-known greedy approaches for subset selection are the submodular function optimization algorithms (Fujishige, 2005; Krause and Golovin, 2014), such as facility location (Frieze, 1974) and entropy function maximization (Sharma et al., 2015). Various coresets strategies have also been designed for specific classes of regression models, such as k-nearest neighbours and naive Bayes (Wei et al., 2015), logistic regression (Guo and Schuurmans, 2007), linear and kernel regression with Gaussian noise (Yu et al., 2006) and support vector machines (Tsang et al., 2005). Such approaches have been designed as active learning strategies and could be developed by exploiting the knowledge of the respective model classes, but do not rely on the models predictions. Assuming the knowledge of learning model may even lead to the development of training set selection strategies that are optimal with respect to some optimality concept, as in the case of linear regression (John and Draper, 1975). Unfortunately, these selection strategies theoretically guarantee benefits only for certain classes of models. In this work, we are interested in passive sampling strategies that are model-agnostic, thus having the potential to benefit multiple classes of regression models rather than just one.

We investigate the benefits of employing the FPS algorithm (Eldar et al., 1994) for training dataset selection. The farthest point sampling is a greedy algorithm that selects elements by attempting to minimize the fill distance of the selected set, which is the maximal distance between the elements in the set of interest and their closest selected element. The work most similar to our is Sener and Savarese (2018), where the authors show that selecting the training set by fill distance minimization can reduce the average classification error on new points for convolutional neural networks (CNNs) with softmax output layers and bounded error function. However, these benefits do not necessarily extend to regression problems, even with simpler Lipschitz algorithms like KRR and FNN, as we illustrate with our experiments. The advantages of using FPS, thus of selecting training sets with a small fill distance, have also been investigated in the context of ML regression. For instance, in Yu and Kim (2010) the authors argue that for regression problems passive sampling strategies, as FPS, are a better choice than active learning techniques. Moreover, in Wu et al. (2019) and Cersonsky et al. (2021), the authors have proposed variations of FPS, and they argue that these can result in more effective training sets. These variations involve selecting the initial point according to a specific strategy rather than randomly, and exploiting the knowledge of labels, when these are known in advance, to obtain subsets that are representative of the whole set in both feature and label spaces. However, these works only demonstrate the advantages of FPS and its variations empirically and do not provide any theoretical analysis to motivate the benefits of using these techniques for regression.

3 Problem definition

We now formally define the problem. We consider a supervised regression problem defined on the feature space $\mathcal{X} \subset \mathbb{R}^d$ and the label space $\mathcal{Y} \subset \mathbb{R}$. We assume the solution of the regression problem to be in a function space $\mathcal{M} := \{f : \mathcal{X} \to \mathcal{Y}\}$, and that for each set of weights $\boldsymbol{w} \in \mathbb{R}^m$ there exists a function in $\mathcal{M}$ associated with it. $\mathcal{M}$ can be interpreted as the space of functions that we can learn by training a given regression approach through the optimization of its weights $\boldsymbol{w} \in \mathbb{R}^m$. Additionally, we consider an error function $l : \mathcal{X} \times \mathcal{Y} \times \mathcal{M} \to \mathbb{R}^+$. The error function takes as input the features of a data point, its label, and a trained regression model and outputs a real value that measures the quality of the prediction of the model for the given data point. The smaller the error, the better the prediction.

Furthermore, we consider a dataset $\mathcal{D} := \{(\boldsymbol{x}_q, y_q)\}_{q=1}^k \subset \mathcal{X} \times \mathcal{Y}$, $k \in \mathbb{N}$, consisting of independent realizations of random variables $(\boldsymbol{X}, Y)$ taking values in $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ with joint probability measure $p_{\mathcal{Z}}$. We study a scenario in which we have only access to the realizations $\{\boldsymbol{x}_q\}_{q=1}^k$, while the labels $\{y_q\}_{q=1}^k$ are unknown, and the goal is to use ML techniques to predict the labels accurately and fast, recovering from data the relation between the random variables $\boldsymbol{X}$ and Y. In supervised ML, we first label a subset $\mathcal{L} := \{(\boldsymbol{x}_{q_j}, y_{q_j})\}_{j=1}^b \subset \mathcal{D}$, $b \ll k$, with $q_j \in \{1, 2, \dots, k\}$ $\forall j$. We then train a regression model $m_{\mathcal{L}} : \mathcal{X} \to \mathcal{Y}$ using a learning algorithm $A(\cdot) : 2^{\mathcal{D}} \to \mathbb{R}^m$ that maps a labelled subset $\mathcal{L} \subset \mathcal{D}$ into weights $\boldsymbol{w} \in \mathbb{R}^m$ determining the learned function $m_{\mathcal{L}} \in \mathcal{M}$ used to predict the labels of the remaining unlabelled points in $\mathcal{U} := \mathcal{D} - \mathcal{L}$. The symbol $2^{\mathcal{D}}$ represents the set of all possible subsets of $\mathcal{D}$. In the following, we renumber the indices $\{q_j\}_{j=1}^b$ associated with the selected set $\mathcal{L}$, and denote them as j, that is, $\mathcal{L} := \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b$. Furthermore, given a set $\mathcal{L} := \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b \subset \mathcal{D}$ we define $\mathcal{L}_{\mathcal{X}} := \{\boldsymbol{x}_j\}_{j=1}^b$ and $\mathcal{L}_{\mathcal{Y}} := \{y_j\}_{j=1}^b$.

In several applications the labelling process is computationally expensive, therefore, given a budget $b \ll k$ of points to label, the goal is to select a subset $\mathcal{L} \subset \mathcal{D}$ with $|\mathcal{L}| = b$

that is most beneficial to the learning process of algorithm $A(\cdot)$. In this work we focus on promoting robustness of the predictions, that is, we want to minimize the maximum expected error of the predictions of the labels obtained with the learned function. Specifically, the problem we want to solve can be expressed as follows:

$$\min_{\substack{\mathcal{L}\subset\mathcal{D},\\ |\mathcal{L}|=b}} \max_{(\boldsymbol{x},y)\in\mathcal{U}} \mathbb{E}[l(\boldsymbol{x}, y, m_{\mathcal{L}})|\boldsymbol{x}], \tag{1}$$

In other words, we aim to select and label a training set $\mathcal{L}$ of cardinality b, so that the maximum expected error associated to a trained regression model $m_{\mathcal{L}}$ evaluated on the unlabelled points is minimized. We focus on model-agnostic training set sampling strategies that have the potential to benefit various learning algorithms. In particular, we do not optimize the data selection process to benefit only an a priori chosen class of learning models.

4 Fill distance minimization by Farthest Point Sampling

Direct computation of the solution to the optimization problem in (1) is not possible as we do not know the labels for the points. To cope with this issue, we derive an upper bound for the minimization objective in (1) that depends linearly on the fill distance of the training set. Afterwards, we describe FPS, which provides a computationally feasible approach to obtain suboptimal solution for minimizing the fill distance.

4.1 Effects of a training set fill distance minimization approach.

First, let us introduce the fill distance, a quantity we can associate with subsets of the pool of data points we wish to label. It can be calculated by considering only the features of the data points.

Definition 1 *Given $\mathcal{D}_{\mathcal{X}} := \{\boldsymbol{x}_q\}_{q=1}^k$ subset of $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{L}_{\mathcal{X}} = \{\boldsymbol{x}_j\}_{j=1}^b \subset \mathcal{D}_{\mathcal{X}}$, the* fill distance *of $\mathcal{L}_{\mathcal{X}}$ in $\mathcal{D}_{\mathcal{X}}$ is defined as*

$$h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}} := \max_{\boldsymbol{x}\in\mathcal{D}_{\mathcal{X}}} \min_{\boldsymbol{x}_j\in\mathcal{L}_{\mathcal{X}}} \|\boldsymbol{x}-\boldsymbol{x}_j\|_2 \tag{2}$$

where $\|\cdot\|_2$ is the L_2-norm. Put differently, we have that any point $\boldsymbol{x} \in \mathcal{D}_{\mathcal{X}}$ has a point $\boldsymbol{x}_j \in \mathcal{L}_{\mathcal{X}}$ not farther away than $h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}}$.

Notice that the fill distance depends on the distance metric we consider in the feature space $\mathcal{X}$. In this work, for simplicity, we consider the L_2-distance, but the following result can be generalized to other distances.

Next, we formulate two assumptions that we use in the theoretical result. The first assumption concerns the data being analyzed and the relationship between features and labels.

Assumption 2 *We assume there exists $\epsilon \geq 0$ such that for each data point $(\boldsymbol{x}_q, y_q) \in \mathcal{D}$ we have that*

$$\mathbb{E}\left[\,|Y-\mathbb{E}[Y|\boldsymbol{x}_q]|\,\middle|\boldsymbol{x}_q\right] := \int_{\mathcal{Y}} \left|y-\mathbb{E}[Y|\boldsymbol{x}_q]\right| p(y|\boldsymbol{x}_q)dy \leq \epsilon, \tag{3}$$

where

$$p(y|\boldsymbol{x}_q) := \frac{p_{\mathcal{Z}}(\boldsymbol{x}_q, y)}{p_{\boldsymbol{X}}(\boldsymbol{x}_q)} \quad \textit{and} \quad p_{\boldsymbol{X}}(\boldsymbol{x}_q) := \int_{\mathcal{Y}} p_{\mathcal{Z}}(\boldsymbol{x}_q, y)dy. \tag{4}$$

We refer to 'ϵ' as the labels uncertainty. Moreover, we assume that

$$\big|\,\mathbb{E}\,[Y|\hat{\boldsymbol{x}}] - \mathbb{E}\,[Y|\tilde{\boldsymbol{x}}]\,\big| \leq \lambda_p \|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}\|_2, \tag{5}$$

$\forall\, \hat{\boldsymbol{x}}, \tilde{\boldsymbol{x}} \in \mathcal{X}$.

Formula (3) states that given a realization $\boldsymbol{X} = \boldsymbol{x}_q$, the average absolute difference between the variable Y and its conditional expectation, taken over the distribution of Y given $\boldsymbol{x}_q$, is bounded by a positive scalar ϵ. In simpler words, given a data point location $\boldsymbol{x}_q \in \mathcal{X}$ in the feature space, its associated label value is not fixed. Instead, it tends to be concentrated in a small region of the label space around its conditional expected value, whose size is determined by the positive scalar ϵ. Formula (3) models those scenarios where the underlying true mapping between the feature and label spaces is either stochastic in nature or deterministic with error fluctuations of magnitude parameterized by ϵ. The Lipschitz continuity in (5) is an assumption on the regularity of the map connecting the feature space $\mathcal{X}$ with the label space $\mathcal{Y}$. It tells us that if two data points have close representations in the feature space, then the conditional expectations of the associated labels are also close, that is, elements closer in $\mathcal{X}$ are more likely to be associated labels close in $\mathcal{Y}$.

The second assumption concerns the error function used to evaluate the performance of the model and the prediction quality of the model on the training set. Firstly, to formalize the notion that the prediction error of a trained model on the training set is bounded. Secondly, to limit our analysis to error functions that exhibit a certain degree of regularity, which also reflects the regularity of the regression model.

Assumption 3 *We assume there exist $\epsilon_{\mathcal{L}} \geq 0$, depending on the labelled set $\mathcal{L} \subset \mathcal{D}$ and the trained regression model $m_{\mathcal{L}}$, such that for each labelled point $(\boldsymbol{x}_j, y_j) \in \mathcal{L}$ we have that*

$$\mathbb{E}[l(\boldsymbol{x}_j, Y, m_{\mathcal{L}})\big|\boldsymbol{x}_j] \leq \epsilon_{\mathcal{L}}. \tag{6}$$

We consider $\epsilon_{\mathcal{L}}$ as the maximum expected prediction error of the trained model $m_{\mathcal{L}}$ on the labelled data $\mathcal{L}$. Moreover, we assume that for any $y \in \mathcal{Y}$ and $\mathcal{L} \subset \mathcal{D}$ the error function $l(\cdot, y, m_{\mathcal{L}})$ is $\lambda_{l_{\mathcal{X}}}$-Lipschitz and that for any $x \in \mathcal{X}$ and $\mathcal{L} \subset \mathcal{D}$, $l(\boldsymbol{x}, \cdot, m_{\mathcal{L}})$ is $\lambda_{l_{\mathcal{Y}}}$-Lipschitz and convex.

With (6) we assume that the expected error on the training set is bounded. Moreover, with the Lipschitz continuity assumptions we limit our study to error functions that show a certain regularity. However, these regularity assumptions on the error function are not too restrictive and are connected with the regularity of the evaluated trained model as we show in Remark 5. For instance, the $\lambda_{l_{\mathcal{Y}}}$-Lipschitz regularity and the convexity in the second argument are verified by all L_p-norm error functions, with $1 \leq p < \infty$.

With that, we formulate the main theoretical result of this work, which is a theorem that provides an upper bound for the optimization objective in (1), depending linearly on the fill distance of the selected training set.

Theorem 4 *Given $\mathcal{D} := \{(\boldsymbol{x}_q, y_q)\}_{q=1}^{k} = \mathcal{U} \sqcup \mathcal{L}$ set of independent realizations of the random variables $(\boldsymbol{X}, Y)$ taking values in $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ with joint probability measure $p_{\mathcal{Z}}$, trained model $m_{\mathcal{L}} \in \mathcal{M}$ and error function $l : \mathcal{X} \times \mathcal{Y} \times \mathcal{M} \to \mathbb{R}^{+}$. If Assumptions 2 and 3 are fulfilled, then we have that*

$$\max_{(\boldsymbol{x},y)\in\mathcal{U}} \mathbb{E}\left[l(\boldsymbol{x}, y, m_{\mathcal{L}})|\boldsymbol{x}\right] \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}} \left(\lambda_{l_{\mathcal{X}}} + \lambda_{l_{\mathcal{Y}}}\lambda_p\right) + \underbrace{\lambda_{l_{\mathcal{Y}}}\epsilon}_{\substack{\textit{labels}\\\textit{uncertainty}}} + \underbrace{\epsilon_{\mathcal{L}}}_{\substack{\textit{max error}\\\textit{training set}}}, \tag{7}$$

where $h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}}$ is the fill distance of $\mathcal{L}_{\mathcal{X}}$ in $\mathcal{D}_{\mathcal{X}}$, ϵ and λ_p are the labels uncertainty and Lipschitz constant from assumption 2, respectively, $\lambda_{l_{\mathcal{X}}}$ and $\lambda_{l_{\mathcal{Y}}}$ are the Lipschitz constants of the error function, and $\epsilon_{\mathcal{L}}$ is the maximum expected error of the trained model predictions on the labelled set $\mathcal{L}$.

Proof First we want to find an upper bound for $\mathbb{E}\left[l(\tilde{\boldsymbol{x}}, Y, m_{\mathcal{L}})|\tilde{\boldsymbol{x}}\right]$ for each $\tilde{\boldsymbol{x}} \in \mathcal{U}_{\mathcal{X}}$. Fixed $\tilde{\boldsymbol{x}} \in \mathcal{U}_{\mathcal{X}}$, by the definition of the fill distance we know there exists $\boldsymbol{x}_j \in \mathcal{L}_{\mathcal{X}}$ such that $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}_j\|_2 \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}}$.

$$\begin{aligned}
\mathbb{E}\left[l(\tilde{\boldsymbol{x}}, Y, m_{\mathcal{L}})|\tilde{\boldsymbol{x}}\right] &= \int_{\mathcal{Y}} l(\tilde{\boldsymbol{x}}, y, m_{\mathcal{L}}) p(y|\tilde{\boldsymbol{x}}) dy \\
&\leq \int_{\mathcal{Y}} \big| l(\tilde{\boldsymbol{x}}, y, m_{\mathcal{L}}) - l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) \big| p(y|\tilde{\boldsymbol{x}}) dy + \int_{\mathcal{Y}} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p(y|\tilde{\boldsymbol{x}}) dy \\
&\leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}} \lambda_{l_{\mathcal{X}}} + \int_{\mathcal{Y}} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p(y|\tilde{\boldsymbol{x}}) dy
\end{aligned} \tag{8}$$

where $\lambda_{l_{\mathcal{X}}}$ is from Assumption 3. The second inequality in (8) follows from the $\lambda_{l_{\mathcal{X}}}$-Lipschitz continuity of the error function. We can bound the remaining term as follows

$$\begin{aligned}
\int_{\mathcal{Y}} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p(y|\tilde{\boldsymbol{x}}) dy &\leq \int_{\mathcal{Y}} \big| l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) - l(\boldsymbol{x}_j, \mathbb{E}\left[Y|\tilde{\boldsymbol{x}}\right], m_{\mathcal{L}}) \big| p(y|\tilde{\boldsymbol{x}}) dy \\
&\quad + \int_{\mathcal{Y}} \big| l(\boldsymbol{x}_j, \mathbb{E}\left[Y|\tilde{\boldsymbol{x}}\right], m_{\mathcal{L}}) - l(\boldsymbol{x}_j, \mathbb{E}\left[Y|\boldsymbol{x}_j\right], m_{\mathcal{L}}) \big| p(y|\tilde{\boldsymbol{x}}) dy \\
&\quad + \int_{\mathcal{Y}} l(\boldsymbol{x}_j, \mathbb{E}\left[Y|\boldsymbol{x}_j\right], m_{\mathcal{L}}) p(y|\tilde{\boldsymbol{x}}) dy \\
&\leq \lambda_{l_{\mathcal{Y}}} \int_{\mathcal{Y}} \big| y - \mathbb{E}\left[Y|\tilde{\boldsymbol{x}}\right] \big| p(y|\tilde{\boldsymbol{x}}) dy \\
&\quad + \lambda_{l_{\mathcal{Y}}} \int_{\mathcal{Y}} \big| \mathbb{E}\left[Y|\tilde{\boldsymbol{x}}\right] - \mathbb{E}\left[Y|\boldsymbol{x}_j\right] \big| p(y|\tilde{\boldsymbol{x}}) dy \\
&\quad + \int_{\mathcal{Y}} \mathbb{E}[l(\boldsymbol{x}_j, Y, m_{\mathcal{L}}) \big| \boldsymbol{x}_j] p(y|\tilde{\boldsymbol{x}}) dy \\
&\leq \lambda_{l_{\mathcal{Y}}}\epsilon + \lambda_{l_{\mathcal{Y}}} \int_{\mathcal{Y}} (\lambda_p h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}}) \, p(y|\tilde{\boldsymbol{x}}) dy + \int_{\mathcal{Y}} \epsilon_{\mathcal{L}} \, p(y|\tilde{\boldsymbol{x}}) dy \\
&\leq \lambda_{l_{\mathcal{Y}}}\epsilon + \lambda_{l_{\mathcal{Y}}}\lambda_p h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}} + \epsilon_{\mathcal{L}}.
\end{aligned} \tag{9}$$

The second inequality follows from the $\lambda_{l_{\mathcal{Y}}}$-Lipschitz continuity of the error function and Jensen's inequality, which is used to obtain the conditional expectation in the integrand of the last term. The third inequality follows from the definition of labels uncertainty, the λ_p-Lipschitz continuity of the conditional expectation of the random variable Y and the assumption that the expected error on the training set is bounded by $\epsilon_{\mathcal{L}}$. The fourth inequality is obtained by taking out the constants from the integrals in the second and third terms and noticing that, from the definition of $p(y|\tilde{\boldsymbol{x}})$ in (4), we have $\int_{\mathcal{Y}} p(y|\tilde{\boldsymbol{x}})dy = 1$. Since the above inequality holds for each $\tilde{\boldsymbol{x}} \in \mathcal{U}_{\mathcal{X}}$, we have that

$$\max_{(\boldsymbol{x},y)\in\mathcal{U}} \mathbb{E}\left[l(\boldsymbol{x}, y, m_{\mathcal{L}})|\boldsymbol{x}\right] \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{D}_{\mathcal{X}}} \left(\lambda_{l_{\mathcal{X}}} + \lambda_{l_{\mathcal{Y}}}\lambda_p\right) + \lambda_{l_{\mathcal{Y}}}\epsilon + \epsilon_{\mathcal{L}}. \tag{10}$$

■

Formula (7) provides an upper bound for the minimization objective in (1) that is linearly dependent on the fill distance of the training set. Note that our derived bound also depends on the labels uncertainty 'ϵ'. In particular, the larger the labels uncertainty, the larger the bound for a fixed training set fill distance. Assuming that the maximum error on the labelled data ($\epsilon_{\mathcal{L}}$) is negligible, the smaller the fill distance, the smaller the bound for the maximum expected approximation error on the unlabelled set, conditional to the unlabelled data locations. Although $\epsilon_{\mathcal{L}}$ is typically considered to be small, its presence in the formula suggests that the maximum expected error on the unlabelled set is also dependent on the maximum error of the predictions on the labelled set used for training, thus, on how well the trained model fits the training data. Additionally, the connection between the bound and the regularity of the map connecting the features and the labels, and the chosen error function are highlighted by the presence of the Lipschitz constants λ_p, $\lambda_{l_{\mathcal{X}}}$ and $\lambda_{l_{\mathcal{Y}}}$ on the right-hand side of (7).

Finally, we remark that if we consider the error function to be the absolute value of the difference between true and predicted labels, Theorem 4 holds for all Lipschitz continuous learning algorithms, such as kernel ridge regression with the Gaussian kernel and feed forward neural networks.

Remark 5 *If the trained model $m_{\mathcal{L}} \in \mathcal{M}$ is $\lambda_{l_{\mathcal{X}}}-$Lipschitz continuous, then also the absolute value error function is $\lambda_{l_{\mathcal{X}}}-$Lipschitz continuous. To see this, fix $y \in \mathcal{Y}$, $\mathcal{L} \subset \mathcal{D}$ and $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$. Then we have*

$$|l(\boldsymbol{x}, y, m_{\mathcal{L}}) - l(\tilde{\boldsymbol{x}}, y, m_{\mathcal{L}})| = \big||m_{\mathcal{L}}(\boldsymbol{x}) - y| - |m_{\mathcal{L}}(\tilde{\boldsymbol{x}}) - y|\big| \leq |m_{\mathcal{L}}(\boldsymbol{x}) - m_{\mathcal{L}}(\tilde{\boldsymbol{x}})|.$$

Moreover, the absolute value error function is always $\lambda_{l_{\mathcal{Y}}}$-Lipschitz with $\lambda_{l_{\mathcal{Y}}} = 1$. As a matter of fact, fixed $\boldsymbol{x} \in \mathcal{X}$, $m_{\mathcal{L}} \in \mathcal{M}$ and $y, \tilde{y} \in \mathcal{Y}$ we have

$$|l(\boldsymbol{x}, y, m_{\mathcal{L}}) - l(\boldsymbol{x}, \tilde{y}, m_{\mathcal{L}})| = \big||m_{\mathcal{L}}(\boldsymbol{x}) - y| - |m_{\mathcal{L}}(\boldsymbol{x}) - \tilde{y}|\big| \leq |y - \tilde{y}|.$$

4.2 Selecting training sets with farthest point sampling

Theorem 4 provides an upper bound for the maximum expected value of the error function on the unlabelled data, conditional to the knowledge of the data features. Our aim is to select a training set by minimizing such a bound. Assuming that the value of the maximum prediction error of the trained regression model on the training set is negligible, we can

Algorithm 1 Farthest Point Sampling (FPS)

```
Input Dataset $\mathcal{D}_{\mathcal{X}} = \{\boldsymbol{x}_q\}_{q=1}^{k} \subset \mathcal{X}$ and data budget $b \in \mathbb{N},\ b \ll k$.
Output Subset $\mathcal{L}_{\mathcal{X}}^{FPS} \subset D_{\mathcal{X}}$ with $|\mathcal{L}_{\mathcal{X}}^{FPS}| = b$.
Choose $\hat{\boldsymbol{x}} \in \mathcal{D}_{\mathcal{X}}$ randomly and set $\mathcal{L}_{\mathcal{X}}^{FPS} = \hat{\boldsymbol{x}}$.
while $|\mathcal{L}_{\mathcal{X}}^{FPS}| < b$ do
    $\bar{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}_q \in \mathcal{D}_{\mathcal{X}}} \min_{\boldsymbol{x}_j \in \mathcal{L}_{\mathcal{X}}^{FPS}} \|\boldsymbol{x}_q - \boldsymbol{x}_j\|_2$.
    $\mathcal{L}_{\mathcal{X}}^{FPS} \leftarrow \mathcal{L}_{\mathcal{X}}^{FPS} \cup \bar{\boldsymbol{x}}$.
end while
```

attempt the minimization of the upper bound in (7) by solving the following minimization problem

$$\min_{\substack{\mathcal{L} \subset \mathcal{D}, \\ |\mathcal{L}| = b}} h_{\mathcal{L}_{\mathcal{X}}, \mathcal{D}_{\mathcal{X}}}, \tag{11}$$

where $\mathcal{D} := \{(\boldsymbol{x}_q, y_q)\}_{q=1}^{k} \subset \mathcal{X} \times \mathcal{Y}$ is the pool of data points we want to label, and $\mathcal{L} := \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{b}$ is the set of labelled points we use for training. The minimization problem in (11) is equivalent to the k-center clustering problem (Har-Peled, 2011). Given a set of points in a metric space, the k-center clustering problem consists of selecting k points, or centers, from the given set so that the maximum distance between a point in the set and its closest center is minimized, i.e., the fill distance of the k centers in the set is minimized. Unfortunately, the k-center clustering problem is NP-Hard (Hochbaum, 1984). However, using farthest point sampling (FPS), described in Algorithm 1, it is possible to obtain sets with fill distance at most a factor of 2 from the minimal fill distance in polynomial time (Har-Peled, 2011). It is worth to note that reducing the factor of approximation below 2 would require solving an NP-hard problem (Hochbaum and Shmoys, 1985).

The FPS can be implemented using $\mathcal{O}(|\mathcal{D}|)$ space and takes $\mathcal{O}(|\mathcal{D}||\mathcal{L}^{FPS}|)$ time (Har-Peled, 2011). Thus, FPS provides a suboptimal solution, but obtaining a better approximation with theoretical guarantees would not be feasible in polynomial time. To give a qualitative understanding of the data efficiency of FPS, with our implementation of the FPS algorithm, it takes approximately 70 seconds[1] to select 1000 points from the training dataset provided within the selection-for-vision DataPerf challenge (Mazumder et al., 2022), consisting of circa 3.3 millions points in $\mathbb{R}^{256}$.

5 Kernel ridge regression with the Gaussian kernel (KRR)

This section focuses on kernel regression models with the Gaussian kernel, a class of regression approaches successfully employed in various applications such as molecular and material sciences (Deringer et al., 2021), or robotics (Deisenroth et al., 2015). Besides considering the Lipschitz continuity of the (absolute) error function with this regression model, we also study how selecting the training set with the FPS increases the model stability for this specific class of regression approaches.

[1] We used a 48-cores CPU with 384 GB RAM.

Kernel ridge regression is a machine learning technique that combines the concepts of kernel methods and ridge regression to perform non-parametric, regularized regression (Deringer et al., 2021). In this work, we use a Gaussian kernel function. Given two data points $\boldsymbol{x}_q, \boldsymbol{x}_r \in \mathcal{X}$, the Gaussian kernel is defined as follows:

$$k(\boldsymbol{x}_q, \boldsymbol{x}_r) := e^{-\gamma \|\boldsymbol{x}_q - \boldsymbol{x}_r\|_2^2}, \tag{12}$$

where $\gamma \in \mathbb{R}^+$ is a kernel hyperparameter to be selected through an optimization process. Provided a training set $\mathcal{L} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b$, the weights $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_b]^T \in \mathbb{R}^b$ of a KRR model are given by the solution of the following minimization problem

$$\boldsymbol{\alpha} = \arg\min_{\bar{\boldsymbol{\alpha}}} \sum_{j=1}^{b} (m_{\mathcal{L}}(\boldsymbol{x}_j) - y_j)^2 + \lambda \bar{\boldsymbol{\alpha}}^T \boldsymbol{K}_{\mathcal{L}}\, \bar{\boldsymbol{\alpha}}. \tag{13}$$

Here, $\boldsymbol{K}_{\mathcal{L}} \in \mathbb{R}^{b,b}$ is the kernel matrix, i.e., $\boldsymbol{K}_{\mathcal{L}}(q,r) = k(\boldsymbol{x}_q, \boldsymbol{x}_r)$, and the parameter $\lambda \in \mathbb{R}^+$ is the so-called regularization parameter that addresses eventual ill-conditioning problems of the matrix $\boldsymbol{K}_{\mathcal{L}}$. The scalars $\{m_{\mathcal{L}}(\boldsymbol{x}_j)\}_{j=1}^b$ are the labels predicted by the KRR method associated with the training data locations $\{\boldsymbol{x}_j\}_{j=1}^b$. The analytic solution to the minimization problem in (13) is given by

$$\boldsymbol{\alpha} = (\boldsymbol{K}_{\mathcal{L}} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y} \tag{14}$$

where $\boldsymbol{y} = [y_1, y_2, \dots, y_b]^T$.

Given the location $\boldsymbol{x} \in \mathcal{X}$ of a new data point, its associated predicted label $y(\boldsymbol{x})$ is defined as follows

$$y(\boldsymbol{x}) := m_{\mathcal{L}}(\boldsymbol{x}) = \sum_{j=1}^{b} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j). \tag{15}$$

5.1 Kernel ridge regression with data selected by FPS

To address the question of the Lipschitz continuity of the KRR with the Gaussian kernel in view of Theorem 4 and Remark 5, we have the following lemma:

Lemma 6 *If the error function is the absolute difference between the true and predicted labels, then the regression function provided by the kernel ridge regression algorithm with the Gaussian kernel is Lipschitz continuous.*

Proof Consider the training set features $\mathcal{L}_{\mathcal{X}} = \{\boldsymbol{x}_j\}_{j=1}^b$ and set of learned weights $\boldsymbol{\alpha}_{\mathcal{L}} := [\alpha_1, \alpha_2, \dots, \alpha_b]^T \in \mathbb{R}^b$ obtained by training the KRR on $\mathcal{L}$. Then, given $\boldsymbol{x} \in \mathcal{X}$ the predicted label $y(\boldsymbol{x})$ provided the KRR approximation function can be computed as follows:

$$y(\boldsymbol{x}) = \sum_{j=1}^{b} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j) = \boldsymbol{\alpha}_{\mathcal{L}}^T \boldsymbol{k}_{\boldsymbol{x}}, \tag{16}$$

where $k(\boldsymbol{x}, \boldsymbol{x}_j) := e^{-\gamma\|\boldsymbol{x}-\boldsymbol{x}_j\|_2^2}$, and $\boldsymbol{k}_{\boldsymbol{x}} := [k(\boldsymbol{x}, \boldsymbol{x}_1), k(\boldsymbol{x}, \boldsymbol{x}_2), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_b)]^T \in \mathbb{R}^b$. Next, considering $\tilde{\boldsymbol{x}}, \hat{\boldsymbol{x}} \in \mathcal{X}$, we have

$$\begin{aligned}
|y(\tilde{\boldsymbol{x}}) - y(\hat{\boldsymbol{x}})| &\leq |\boldsymbol{\alpha}_{\mathcal{L}}^T \boldsymbol{k}_{\tilde{\boldsymbol{x}}} - \boldsymbol{\alpha}_{\mathcal{L}}^T \boldsymbol{k}_{\hat{\boldsymbol{x}}}| \\
&\leq \|\boldsymbol{\alpha}_{\mathcal{L}}\|_2 \|\boldsymbol{k}_{\tilde{\boldsymbol{x}}} - \boldsymbol{k}_{\hat{\boldsymbol{x}}}\|_2 \\
&= \|\boldsymbol{\alpha}_{\mathcal{L}}\|_2 \sqrt{\sum_{j=1}^{b} \left(e^{-\gamma\|\tilde{\boldsymbol{x}}-\boldsymbol{x}_j\|_2^2} - e^{-\gamma\|\hat{\boldsymbol{x}}-\boldsymbol{x}_j\|_2^2}\right)^2} \\
&\leq \|\boldsymbol{\alpha}_{\mathcal{L}}\|_2 \sqrt{b} \lambda_k \|\tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}\|_2,
\end{aligned}$$

where λ_k is the Lipschitz constant of the function $e^{-\gamma r^2}$, $r \in \mathbb{R}^+$. ■

5.2 Increased numerical stability of Gaussian kernel regression with FPS

Numerical stability in a regression approach is a key factor in ensuring the robustness of the learning algorithm with respect to noise and therefore its reliability. A standard criterion for measuring the numerical stability in case of kernel regression is the condition number of the kernel matrix, $\mathbf{K}_{\mathcal{L}} \in \mathbb{R}^{b,b}$. In the specific case of a Gaussian kernel we have that $\mathbf{K}_{\mathcal{L}}(i,j) := e^{-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2}$, $\gamma \in \mathbb{R}^+$. The condition number of a matrix is defined as

$$cond(\mathbf{K}_{\mathcal{L}}) := \|\mathbf{K}_{\mathcal{L}}\|_2 \|\mathbf{K}_{\mathcal{L}}^{-1}\|_2 = \frac{\lambda_{max}(\mathbf{K}_{\mathcal{L}})}{\lambda_{min}(\mathbf{K}_{\mathcal{L}})}, \tag{17}$$

where $\lambda_{max}(\mathbf{K}_{\mathcal{L}})$ and $\lambda_{min}(\mathbf{K}_{\mathcal{L}})$ are the largest and smallest eigenvalues of $\mathbf{K}_{\mathcal{L}}$, respectively. The smaller the condition number, the more numerically stable the algorithm. For high condition numbers, the numerical computations involving the kernel matrix can suffer from amplification of rounding errors and loss of precision that can lead to numerical instability when performing operations like matrix inversion or solving linear systems involving the kernel matrix. Such phenomena may also lead to instability of the predictions as small variations in the input may lead to significant variations in the output.

To increase the model stability we aim to select a training set that leads to a kernel matrix with a small condition number (17). From the literature (Wendland, 2004), we know that the largest eigenvalue of a kernel matrix is mainly dependent on the number of points we consider and not on how we choose them. In particular, the value of the largest eigenvalue can be bounded as follows

$$\lambda_{max}(\mathbf{K}_{\mathcal{L}}) \leq b \max_{q,r=1,\ldots,b} |\mathbf{K}_{\mathcal{L}}(q,r)|. \tag{18}$$

Thus, the maximum eigenvalue is bounded by a quantity that depends linearly on the number of training samples times the maximal entry of the kernel matrix. Since we are considering Gaussian kernels, the maximal entry of the kernel is bounded. Consequently, the value of the maximal eigenvalue grows at most as fast as the number of points we select, independently of how we choose them.

On the contrary, the value of the smallest eigenvalue is strongly dependent on how we choose the training points. To study that, we use the separation distance, a quantity we can associate to subsets of our pool of unlabelled data points.

Definition 7 *Given set $\mathcal{L}_\mathcal{X} := \{\boldsymbol{x}_j\}_{j=1}^b \in \mathbb{R}^d$, the* separation distance *of the points in $\mathcal{L}_\mathcal{X}$ defined as*

$$s_{\mathcal{L}_\mathcal{X}} := \frac{1}{2} \min_{\substack{\boldsymbol{x}_q, \boldsymbol{x}_r \in \mathcal{L}_\mathcal{X} \\ q \neq r}} \|\boldsymbol{x}_q - \boldsymbol{x}_r\|_2.$$

In words, the separation distance is half the minimal distance between two points in $\mathcal{L}_\mathcal{X}$. Given a training set $\mathcal{L} \subset \mathcal{D}$ we define $s_{\mathcal{L}_\mathcal{X}}$ to be its separation distance.

With the concept of separation distance in mind, we observe that the value of the smallest eigenvalue of the Gaussian kernel matrix can be bounded from below as (Wendland, 2004)

$$\lambda_{min}(\mathbf{K}_\mathcal{L}) \geq C_d \left(\sqrt{2\gamma}\right)^{-d} e^{\frac{-40.71 d^2}{(s^2_{\mathcal{L}_\mathcal{X}} \gamma)}} s_{\mathcal{L}_\mathcal{X}}^{-d}, \tag{19}$$

where $d \in \mathbb{N}$ is the training data dimension, which is fixed, $\gamma \in \mathbb{R}^+$ is the Gaussian kernel hyperparameter, representing the width of the Gaussian, and $s_{\mathcal{L}_\mathcal{X}} \in \mathbb{R}^+$ is the training set separation distance. It is important to notice that the lower bound of the smallest eigenvalue decreases exponentially as the separation distance of the selected set decreases. Consequently, given two training sets of the same size, a small difference in their separation distance may lead to a large difference between the smallest eigenvalue of their corresponding kernel matrices, thus also in condition number and model stability.

Therefore, in order to increase the model stability of the kernel regression approach, we aim to select a training set that solves the following NP optimization problem

$$\max_{\substack{\mathcal{L} \subset \mathcal{D} \\ |\mathcal{L}| = b}} s_{\mathcal{L}_\mathcal{X}}. \tag{20}$$

Interestingly, the FPS provides sets with separation distance at most a factor of 2 from the maximal separation distance (Eldar et al., 1994). Moreover, to obtain an approximation factor better than 2, an NP problem must be solved. Thus, if we restrict the spectrum of the selection strategies of interest to those that run in polynomial time, as it is the case when we work with large datasets, the FPS provides optimal solutions to both the problems in (11) and (20). Consequently, when we consider kernel regression approaches with the Gaussian kernel, selecting the training set with the FPS leads to more robust and stable models.

6 Experimental results

We investigate the effects of minimizing the training set fill distance on regression tasks from quantum chemistry, where molecular properties are predicted on the QM7, QM8 and QM9 datasets. In particular, we study the performance of FPS in comparison to several sampling baselines while using two machine learning models for prediction, KRR and FNN.

6.1 Datasets

QM7 (Blum and Reymond, 2009; Rupp et al., 2012) is a benchmark dataset in quantum chemistry, consisting of 7165 small organic molecules with up to 23 atoms including 7 heavy atoms: C, N, O and S. It includes information such as the Cartesian coordinates and the atomization energy of the molecules. We use QM7 for a regression task, where the feature vector for a molecule is the Coulomb matrix (Rupp et al., 2012). The Coulomb matrix is defined as

$$\boldsymbol{C}_{i,j} = \begin{cases} \frac{1}{2} z_i^{2.4} & \text{if } i = j \\ \frac{z_i z_j}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2} & \text{if } i \neq j \end{cases} \tag{21}$$

where z_i is the nuclear charge of the i-th atom and $\boldsymbol{r}_i$ is its position. In the case of QM7 each molecule is thereby represented as an element in $\mathbb{R}^{529}$, and the label value to predict is the atomization energy, a scalar value describing amount of energy in electronvolt (eV) required to completely separate all the atoms in a molecule into individual gas-phase atoms.

QM8 (Ruddigkeit et al., 2012; Ramakrishnan et al., 2015) is a curated collection of 21,786 organic molecules with up to 8 heavy atoms (C, N, O, and F). For each of the molecules it provides the SMILES representation (Weininger, 1988) together with various molecular properties, such as the lowest two singlet transition energies and their oscillator strength. These molecular properties have been computed considering different approaches. In this study we consider those values computed with hybrid exchange correlation functional PBE0. To generate the molecular descriptors we employ Mordred (Moriwaki et al., 2018), a publicly available library that exploits the molecules' topological information encoded in the SMILES strings to provide 1826 physical and chemical features. To work with a more compact representation, we remove 530 features for which the values across the dataset have zero variance. Thus, each molecule in QM8 is represented by a vector in $\mathbb{R}^{1296}$. Furthermore, we normalize the features provided by the Mordred library, to scale them independently in the interval (0, 1). The label value to predict in the regression task is the lowest singlet transition energy (E1), measured in eV, describing the energy difference between the ground state and the lowest excited state in a molecule when both states have singlet spin multiplicity. It is an important property in understanding the electronic behavior of molecules.

QM9 (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014) is a publicly available quantum chemistry dataset containing the properties of 133,885 small organic molecules with up to nine heavy atoms (C, N, O, F). QM9 is frequently used for developing and testing machine learning models for predicting molecular properties and for exploring the chemical space (Faber et al., 2017; Ramakrishnan and von Lilienfeld, 2017; Pronobis et al., 2018). QM9 contains the SMILES representation (Weininger, 1988) of the relaxed molecules, as well as their geometric configurations and 19 physical and chemical properties. In order to ensure the integrity of the dataset, we have excluded all 3054 molecules that did not pass the consistency test proposed by Ramakrishnan et al. (2014). Additionally, we have removed the 612 compounds that could not be interpreted by the RDKit package (Landrum, 2012). Furthermore, in order to ensure the uniqueness of data points, we have excluded 17 molecules that had SMILES representations that were identical to those of other molecules in the dataset. Following this preprocessing procedure, we obtained a smaller version of the QM9 dataset comprising 130202 molecules. The molecular representation we employ is based on the Mordred (Moriwaki et al., 2018) library, as for the QM8 dataset, with the difference that in this case we do not normalize the features, to show that our results are independent of the normalization. To work with a more compact representation, we remove 519 features for which the values across the dataset have zero variance. Thus, each molecule in QM9 dataset is represented by a vector in $\mathbb{R}^{1307}$. The label value to predict is the HOMO-LUMO energy, measured in eV, describing the difference between the highest occupied (HOMO) and the lowest unoccupied (LUMO) molecular orbital energies. It is a useful quantity for examining the molecules kinetic stability.

6.2 Baseline sampling strategies

We compare the effects of minimizing the training set fill distance through the FPS algorithm with three coresets benchmark sampling strategies. Specifically, we consider random sampling (RDM), the facility location algorithm and k-medoids++. Random sampling (RDM) is considered the natural benchmark for all the other coreset sampling strategies (Feldman, 2019), and consists of choosing the points to label and use for training uniformly at random from the available pool of data points. Facility location (Frieze, 1974) is a greedy algorithm that aims at minimizing the sum of the distances between the points in the pool and their closest selected element. k-medoids++ (Mannor et al., 2011) is a variant of the k-means++ (Arthur and Vassilvitskii, 2007), that partitions the data points into k clusters and, for each cluster, selects one data point as the cluster center by minimizing the distance between points labelled to be in a cluster and the point designated as the center of that cluster. Both, facility location and k-medoids++, attempt to minimize a sum of pairwise distances. However, the fundamental difference is that facility location is a greedy technique, while k-medoids++ is based on a segmentation of the data points into clusters.

6.3 Regression models

In this work we use ML regression models that have been utilized in previous works for molecular property prediction tasks. Specifically, we consider kernel ridge regression with the Gaussian kernel (KRR) (Stuke et al., 2019; Deringer et al., 2021) and feed forward neural networks (FNNs) (Pinheiro et al., 2020). KRR and FNN are of interest to us because of their Lipschitz continuity, which, from Remark 5, we know is a required property to validate our theoretical analysis.

We already described KRR in Section 5 and showed the Lipschitz continuity. The hyperparameters γ and λ are optimized through the following process: first we perform a cross-validation grid search to find the best hyperparameter for each training set size using subsets obtained by random sampling. Next, the average of the best parameters pair for each training set size is used to build the final model. The KRR hyperparameters are varied on a grid of 12 points between 10^{-14} and 10^{-2}. Note that, we do not use an optimal set of hyperparameters for each selection strategy and training set size. This decision is made because we aim to analyze the qualitative behaviour of a fixed model, where the only variable affecting the quality of the predictions is the selected training set.

Feed-forward neural networks (Goodfellow et al., 2016) (FNNs) are probably the simplest deep neural networks. Given $\boldsymbol{x} \in \mathcal{X}$ the predicted label, $y(\boldsymbol{x})$, provided by a FNN, with $l \in \mathbb{N}$ layers, can be expressed as the output of a composition of functions, that is,

$$y(\boldsymbol{x}) := \phi_l \circ \sigma_l \circ \phi_{l-1} \circ \sigma_{l-1} \circ \cdots \circ \phi_1(\boldsymbol{x}), \tag{22}$$

where the ϕ_i are affine linear functions or pooling operations and the σ_i are nonlinear activation functions. Following along (Pinheiro et al., 2020), we set $l = 3$, consider only ReLu activation functions and define

$$\phi_i(\boldsymbol{x}) = \boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{b}_i \tag{23}$$

where the weight matrices $\boldsymbol{W}_i$ and the biases $\boldsymbol{b}_i$ are learned by minimizing the mean absolute error between the true and predicted labels of the data points in the training set. The Lipschitz continuity of FNN and other more advanced neural networks has been shown in the literature (Scaman and Virmaux, 2018; Gouk et al., 2020).

6.4 Evaluation metrics

We consider two metrics to evaluate the performance of the ML methods used for the regression tasks: Maximum Absolute Error (MAXAE) and Mean Absolute Error (MAE). The MAXAE is the maximum absolute difference between the true target values $\{y_i\}_{i=1}^n$ and the predicted values $\{\tilde{y}_i\}_{i=1}^n$, that is,

$$\text{MAXAE} := \max_{1\leq i\leq n} |y_i - \tilde{y}_i|, \tag{24}$$

where n is the number of unlabelled data points in the analyzed data pool. The MAE is calculated by averaging the absolute differences between the predicted values and the true target values, that is,

$$\text{MAE} := \frac{1}{n}\sum_{i=1}^{n} |y_i - \tilde{y}_i|. \tag{25}$$

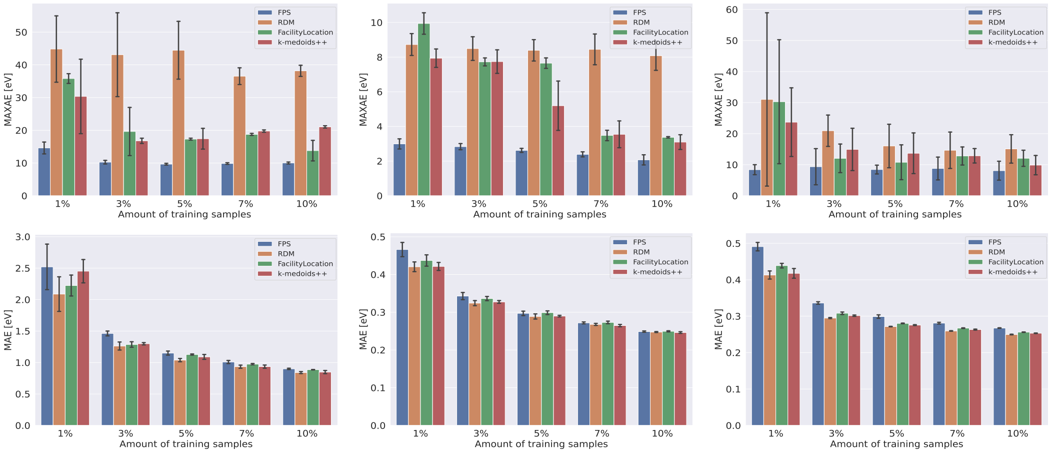
Furthermore, to evaluate the stability of the KRR approach we also consider the condition number of the kernel obtained from the training data defined as in (17).

6.5 Numerical Results

The experiments we perform involve testing the predictive accuracy of each trained model on all data points not used for training, in terms of the MAXAE and MAE. For each sampling strategy, we construct multiple training sets consisting of different amounts of samples. For each sampling strategy and training set size, the training set selection process is independently run five times. In the case of RDM, points are independently and uniformly selected at each run, while for the other sampling techniques, the initial point to initialize is randomly selected at each run. Therefore, for each selection strategy and training set size, each analyzed model is independently trained and tested five times. The reported test results are the average of the five runs. We also plot error bands, which, unless otherwise specified, represent the standard deviation of the results. We remark that the final goal of our experiments is to empirically show the benefits of using FPS compared to other model-agnostic state-of-the-art sampling approaches. We do not make any claims on the general prediction quality of the employed models on any of the studied datasets.

6.5.1 Molecular property prediction on QM7, QM8 and QM9 datasets

Fig. 1 and Fig. 2 show the results for the regression tasks on the QM7, QM8 and QM9 datasets using KRR and FNN, respectively. The graphs on the top rows of Fig. 1 and Fig. 2



(a) KRR on QM7

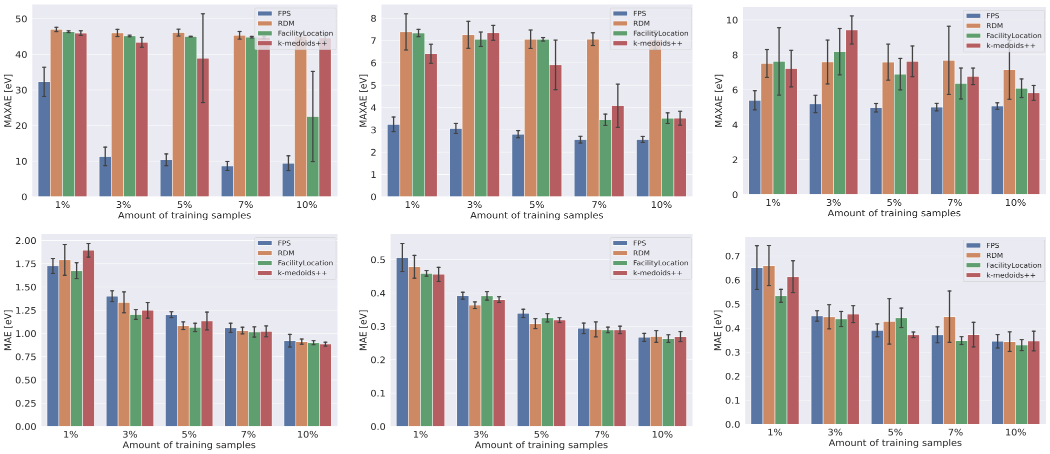
(b) KRR on QM8

(c) KRR on QM9

Figure 1: Results for regression tasks on QM7, QM8 and QM9 using KRR trained on sets of various sizes, expressed as a percentage of the available data points, and selected with different sampling strategies. MAXAE (top row) and MAE (bottom row) are shown for each training set size and sampling approach.

illustrate the maximum error of the predictions on the unlabelled points. The results suggest that, independently of the dataset and the regression model employed for the regression task, selecting the training set by fill distance minimization using FPS, we can perform better than the other baselines in terms of the maximum error of the predictions.

The graphs on the bottom row of Fig. 1 and Fig. 2 show the MAE of the predictions on the QM7, QM8 and QM9 datasets for KRR and FNN, respectively. These graphs indicate that selecting training sets with FPS doesn't drastically reduce the MAE of the predictions on the unlabelled points with respect to the baselines, independently of the dataset and regression model. On the contrary, we observe examples where FPS performs worse than one of the baselines, e.g., with the FNN on the QM7, QM8 and QM9 when trained with 5% of the available data points. These experiments suggest that, contrary to what has been shown for classification (Sener and Savarese, 2018), selecting training sets by fill distance minimization does not provide any significant advantage compared to the baselines in terms of the average error. This marks a fundamental difference between regression and classification tasks regarding the benefits of reducing the training set fill distance. The graphs on the top row of Fig. 3 show the condition number of the regularized kernel matrices generated during training of the KRR approach and used to calculate the regression parameters as shown in (14). For the QM9, the condition number appears not to be affected by the training dataset choice, while for the QM7 and QM8, choosing training sets with FPS reduces the condition number of the regularized kernel, particularly in the low data regime, leading to improved stability of the learned model as discussed in Section 5.2. We remark that the graphs in the top row of Fig. 3 depict the condition number of the regularized kernel matrix $\boldsymbol{K}_{\mathcal{L}} + \lambda \boldsymbol{I}$, where $\boldsymbol{K}_{\mathcal{L}}$ is the Gaussian kernel matrix built from the training data and $\lambda \boldsymbol{I}$ is the regularization term, introduced in (13), used to address ill-conditioning problems. The hyperparameter λ has been chosen following a procedure based



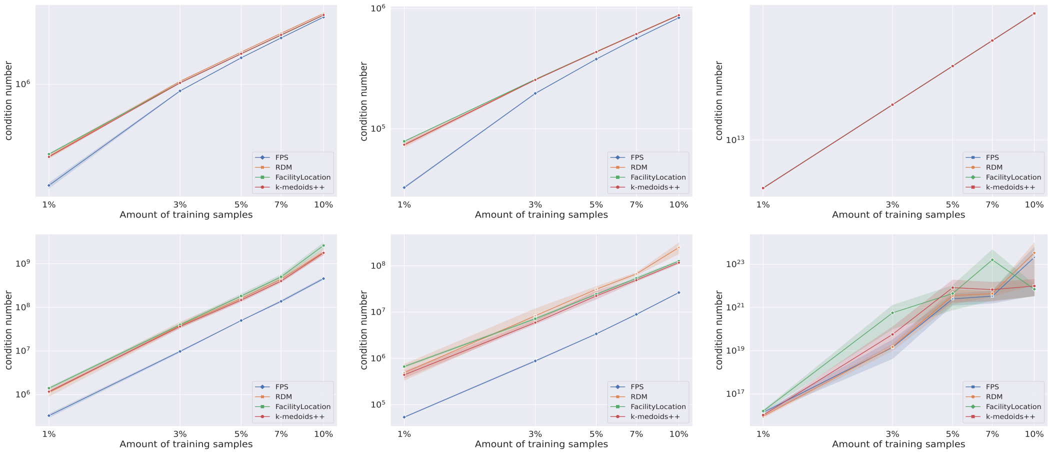
(a) FNN on QM7 (b) FNN on QM8 (c) FNN on QM9

Figure 2: Results regression tasks on QM7, QM8 and QM9 using FNN trained on sets of various sizes, expressed as a percentage of the available data points, and selected with different sampling strategies. MAXAE (top row) and MAE (bottom row) of the predictions are shown for each training set size and sampling approach.

on cross-validation on randomly selected subsets of the available data pool, as explained in Subsection 6.3. The values of λ we employed are $1.9 \cdot 10^{-4}$, $2.2 \cdot 10^{-3}$ and $1.5 \cdot 10^{-11}$ for the QM7, QM8 and QM9, respectively. The bottom row of Fig. 3 illustrates the condition numbers of the non-regularized kernels and shows that, if no regularization is applied, for the QM7 and the QM8 there is an exponential difference between the condition numbers of the matrices obtained with the FPS and those obtained using the benchmark strategies, as expected from (19). As for the QM9, we still see a lower condition number when using the FPS in the low data limit, until 7% of the data is employed for training. Notice that for the QM9, the magnitude of the condition number is significantly higher than for the other datasets due to the larger size of the kernel matrix.

6.5.2 Empirical analysis and discussion

Interestingly, with FPS, the MAXAE converges fast to a plateau value for all datasets and regression models (Fig. 1- 2). Differently, with the baseline approaches, the MAXAE has much larger values in the low data regime and tends to decrease gradually as the size of the training sets increases. It is important to notice that, these trends of the MAXAE of the predictions are directly correlated with the fill distances of the respective labelled sets used for training, illustrated in Fig. 4a. From Fig. 4a it can be clearly seen that independently of the dataset considered, with FPS, the fill distances are consistently lower even for small data budgets, while with the benchmarks, the fill distances are much larger in the low data regime and gradually decrease as the size of the training set increases. These observations indicate that the training set fill distance is directly correlated with the maximum error of the predictions on the unlabelled set. Consequently, by minimizing the training set fill distance, we can drastically reduce the MAXAE of the predictions. Nevertheless, our theoretical analysis shows that the training set fill distance is only linked to the maximum



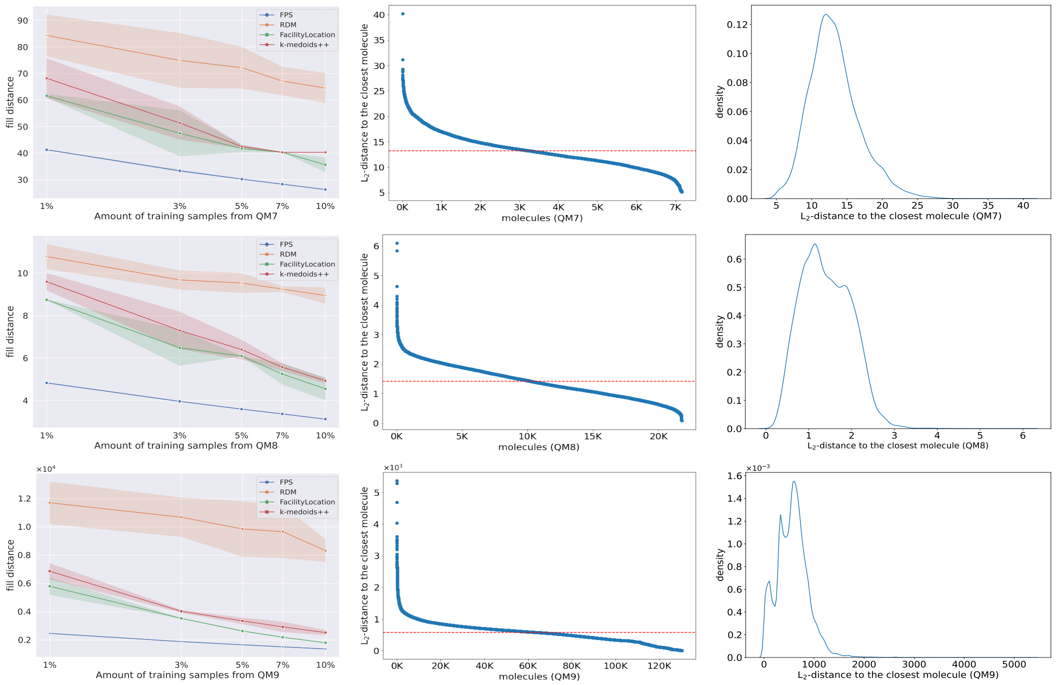
(a) QM7 (b) QM8 (c) QM9

Figure 3: Condition number of the regularized (top row) and non-regularized (bottom row) Gaussian kernels are shown for each dataset, training set size and sampling approach. The graphs are on log-log scale and the error bands represent the confidence interval over five independent runs of the experiments.

expected value of the error function computed on the unlabelled points. Moreover, this bound also depends on other quantities we may not know or that we cannot compute a priori. Namely, the labels uncertainty and the maximum prediction error on the training set, quantifying how well the trained regression model fits the training data. Thus, we believe that the training set fill distance should not be considered as the only parameter to obtain an a priori quantitative evaluation of the MAXAE of the predictions, but as a qualitative indicator of the model robustness that, if minimized, leads to a substantial reduction of the MAXAE.

In addition to our theoretical motivation for the effectiveness of FPS in reducing the MAXAE of the predictions, we now aim to provide a more empirical motivation for the effectiveness of FPS.

In our view, the effectiveness of FPS is also due to its ability to sample, even for small training sets sizes, those points that are at the tails of the data distribution and that are convenient to label, as the predictive accuracy of the learning methods on those points would be limited due to the lack of data information in the portions of the feature space where data points are more sparsely distributed. To see this empirically, let us first consider Fig. 4b and Fig. 4c, showing for each molecule the Euclidean distance to the respective closest molecule and the density of such distances, respectively, for the QM7, QM8 and QM9 datasets. Fig. 4b shows that, in all the analyzed datasets, there are “isolated” molecules for which the Euclidean distance to the nearest molecule is more than twice the average distance between the molecules in the dataset and their nearest neighbour, represented by the red line in the graphs. Fig. 4c, representing the density distribution of the distances of the molecules to their closest data point, tells us that the “isolated” molecules are only a very small portion of the dataset and, therefore, represent the tail of the data distribution. We now see that FPS, contrary to the other baselines, can effectively sample the isolated



(a) Training sets fill distances.

(b) Distances to the nearest neighbour.

(c) Density of distances to the nearest neighbour.

Figure 4: (a) Fill distances of the selected training sets. (b) Euclidean distances to the nearest neighbour and (c) density of such distances for molecules in the QM7 (top row), QM8 (middle row) and QM9 (bottom row). In (b) the red lines are the average distances between the molecules in the datasets and their nearest neighbour and the molecules are sequentially numbered such that the distances decrease in magnitude as the associated molecule numbers increase.

molecules even for a low training data budget. Fig. 5 highlights the Euclidean distances to the closest neighbour for molecules selected with FPS, and the other baseline strategies, from all the analyzed datasets. The size of the selected sets is 1% of the available data points. Specifically, we are analyzing the same elements selected in the lowest training data budget we considered for the regression tasks in Fig. 1 and Fig. 2. Fig. 5 clearly illustrates that, independently of the dataset, FPS selects points across the whole density spectrum. On the contrary, the baseline methods mainly sample points that have a closer nearest neighbour and that are nearer to the center of the data distribution (Fig. 4c). Our hypothesis that selecting isolated molecules is beneficial in terms of the MAXAE reduction is also supported by our theoretical analysis. From Theorem 4, we know that a sampling strategy that aims to reduce the maximum error of the predictions should minimize the fill distance of the training set. Thus, it should include the isolated molecules in the training set, as their distance to the nearest neighbour is much larger than the average.

Our empirical analysis indicates that using FPS can be advantageous in the low training data budget, as it allows including early in the sampling process the “isolated” molecules. But, once the data points at the tails of the data distribution have been included, we believe

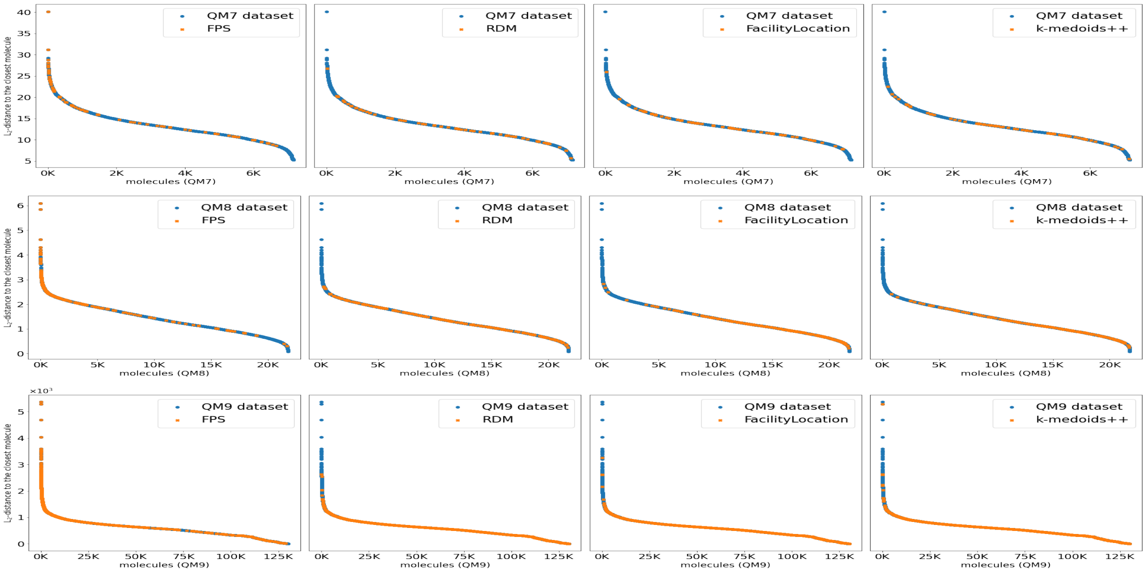
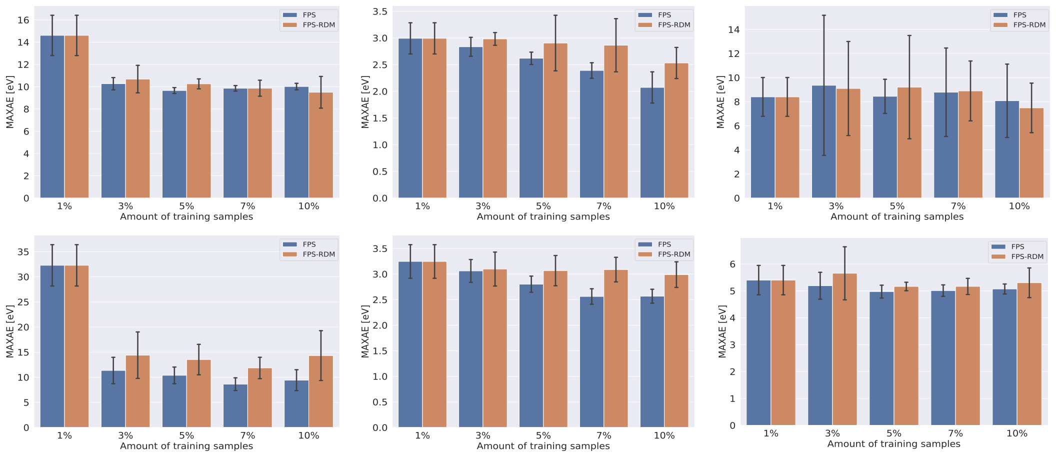


Figure 5: In blue, the Euclidean distances to the nearest neighbour for molecules in the QM7 (top row), QM8 (middle row) and QM9 (bottom row). In orange are highlighted the molecules selected with FPS and the other baselines. For each dataset we selected 1% of available data points.

that there may be more convenient sampling strategies than FPS to select points at the center of the distribution, where more information is available. To empirically support the hypothesis that the FPS is mostly beneficial in the low data limit, Fig. 6 illustrates the MAXAE of the predictions on the QM7, QM8 and QM9, for the KRR and FNN trained on sets selected with the FPS and on sets selected initially with the FPS, the first 2%, and then selected randomly. The figure clearly illustrates that after the FPS has been employed to sample the first 2% of the dataset, the MAXAE of the predictions does not tend to decrease or increase dramatically for larger training set sizes, even if the later samples are selected randomly, independently of the datasets and regression model considered. This fact further suggests that the FPS is mainly beneficial in the low data limit and is strongly connected with the ability of this sampling strategy to select samples at the tail of the data distribution.

6.5.3 Importance of the data assumptions

We now highlight the importance of the data assumptions in ensuring that a fill distance minimization strategy leads to a significant reduction of the MAXAE, in correspondence to the theoretical result proposed in Theorem 4. The focus is on Assumption 2, Formula (5), indicating that if two data points have close representations in the feature space, then the conditional expectations of the associated labels are also close. Simply put, this assumption states that if two data points have similar features, their labels are more likely to be similar as well. Therefore, we expect the pairwise distances in the feature and label space to be directly correlated for the experiments to ensure consistency with the theory.



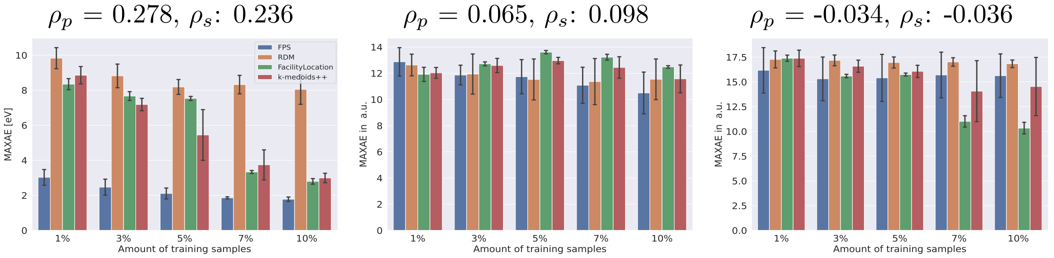
(a) QM7 (b) QM8 (c) QM9

Figure 6: Results for regression tasks on QM7, QM8 and QM9 using KRR (top row) and FNN (bottom row) trained on sets of various sizes and selected with the FPS and with the FPS combined with random sampling after 2% of the available data have been selected. The graphs illustrate the MAXAE of the predictions for each training set size and sampling approach.

One approach to test this hypothesis on a given dataset is to calculate the Euclidean distance in the feature and label spaces for each pair of points and then calculate Pearson's (ρ_p) or Spearman's (ρ_s) correlation coefficient (Boslaugh, 2008) to assess the strength and direction of the correlation between the pairwise distances. These coefficients measure how closely correlated two quantities are, with values ranging from -1 to 1. Pearson's coefficient, captures linear relationships between variables, whereas Spearman's coefficient, measures monotonic relationships regardless of linearity. A positive value indicates a positive correlation, while a negative value indicates a negative correlation. Following along Schober et al. (2018), we define the correlation between the two analyzed quantities to be negligible if the considered correlation coefficient ρ is such that $|\rho| \leq 0.1$, otherwise we consider the correlation positive or negative, depending on the sign of ρ.

We test our hypothesis on the data assumption for the experiments analyzed in 6.5.1 and illustrated in Fig. 1 and Fig. 2. For completeness, we consider both Pearson's (ρ_p) and Spearman's (ρ_s) coefficient. The computed coefficients are 0.149, 0.216, and 0.272 for ρ_p, and 0.281, 0.189, and 0.216 for ρ_s, for QM7, QM8, and QM9, respectively. These numbers indicate that in all experiments where the fill distance minimization approach is successful in significantly reducing the maximum prediction error, there is a positive correlation between the pairwise distances of the data features and labels.

Moreover, we also want to show that if the correlation between the pairwise distances in the feature and label space is negligible, the fill distance minimization approach may not lead to a significant reduction in the maximum prediction error. To illustrate this, we perform additional experiments on the QM8 dataset. In these experiments, we examine various labels not yet considered in this work while considering the same data features we previously used. The labels we now consider are the second singlet transition energy



(a) second singlet transition energy (E2)

(b) first oscillator strength (f1)

(c) second oscillator strength (f2)

Figure 7: Results for regression tasks on QM8 considering three different labels: (a) second singlet transition energy, (b) first oscillator strength, (c) second oscillator strength. Regression performed using KRR trained on sets of various sizes and selected with different sampling strategies. The graphs illustrate the MAXAE of the predictions and the legend in (a) applies to (b) and (c) as well. ρ_p and ρ_s are Pearson's and Spearman's correlation coefficients of the data points pairwise distance in the feature and label space.

(E2), measured in eV, and the first and second oscillator strengths (f1 and f2), measured in atomic units (a.u.). Our computations reveal a Pearson's and Spearman's correlation coefficient of 0.278 and 0.236 for E2, respectively. As for correlations with f1 and f2, Pearson's coefficients are 0.065 and -0.034, respectively, while Spearman's coefficients are 0.098 and -0.036, accordingly. These results suggest a positive correlation between pairwise distances in the feature and label space when considering E2 as the label value, but negligible correlations for f1 and f2. This rejection of our hypothesis for f1 and f2 indicates that our initial assumptions about the data properties may not hold true when considering these two labels. Fig. 7 shows the results for the regression tasks on the QM8 dataset considering E2, f1 and f2 as labels, and using the KRR as regression model. Specifically, Fig. 7b and 7c illustrate the MAXAE of the predictions for the regression tasks with f1 and f2, respectively, and suggest that selecting the training set by fill distance minimization using FPS, does not lead to a significant reduction in the maximum prediction error when the correlation between the pairwise distances in the feature and label space is negligible. On the contrary, Fig. 7a, illustrating the results on the E2 regression task, provides further evidence that the fill distance minimization approach is effective when the correlation is positive, in correspondence to our hypothesis on the data assumption. It is important to note that for the case of the QM8 dataset with f1 or f2 as labels, where the correlation between pairwise distances in the features and label space is negligible, selecting training sets by fill distance minimization approach with the FPS is either comparable or better than randomly choosing the points in terms of the MAXAE of the predictions. Moreover, it is also important to mention that, no benchmark approach can consistently perform better than FPS. For instance, the facility location approach performs best on the f2 regression task for training set sizes of 7% and 10%, but is the worst performing on the f1 regression task for all training set sizes other than 1%.

7 Conclusion and Future work

We study the effects of minimizing the training set fill distance for Lipschitz continuous regression models. Our numerical results have shown that, under the given data assumption, using FPS to select training sets by fill distance minimization increases the robustness of the models by significantly reducing the prediction maximum error, in correspondence to our theoretical motivation. Furthermore, we have shown theoretically and empirically that, if we consider Gaussian kernel regression models, selecting training sets with the FPS also leads to increased model stability. Additionally, we have seen that FPS is particularly advantageous with low training data budgets and argued that there may be more convenient sampling strategies than FPS to select larger amounts of points and improve the average quality of the predictions of a regression model as well.

Based on these remarks, two questions naturally arise: Firstly, how to determine a priori the minimal amount of points to be selected with the FPS? Secondly, how can we modify the FPS to select training sets that can also reduce the average prediction error of a regression model on the data distribution? One possible solution to address the second question is to modify FPS by considering weighted distances. We propose to thereby take the distribution of the data during the sampling process into account, giving higher importance to points in regions of the feature space with higher data density.

Broader Impact Statement

Minimizing the training set fill distance can be highly beneficial in applications where traditional labelling methods, such as numerical simulations or laboratory experiments, are computationally intensive, time-consuming, or costly, such as in the field of molecular property computations. In such applications, ML regression models are used to predict the unknown labels of data points quickly. However, their accuracy is highly dependent on the quality of the training data. Therefore, careful selection of the training set is crucial to ensure accurate and robust predictions for new points. Our research on minimizing the training set fill distance can be used to identify training sets that have the potential to improve prediction robustness across a wide range of regression models and tasks. This approach prevents the wastage of expensive labelling resources on subsets that may only benefit a particular learning model or task.

Acknowledgments and Disclosure of Funding

This work was supported by the German Federal Ministry of Education and Research (BMBF) within the project MaGriDo (Mathematics for Machine Learning Methods for Graph-Based Data with Integrated Domain Knowledge), FKZ 05M20PDB.

References

David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

L. C. Blum and J.-L. Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.*, 131:8732, 2009.

Sarah Boslaugh. *Statistics in a nutshell.* O'Reilly, 2008. ISBN 9780596510497.

Rose K Cersonsky, Benjamin A Helfrecht, Edgar A Engel, Sergei Kliavinek, and Michele Ceriotti. Improving sample and feature selection with principal covariates regression. *Machine Learning: Science and Technology*, 2(3):035038, jul 2021. doi: 10.1088/2632-2153/abfe7c.

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, feb 2015. doi: 10.1109/tpami.2013.218.

Volker L. Deringer, Albert P. Bartók, Noam Bernstein, David M. Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, aug 2021. doi: 10.1021/acs.chemrev.1c00022.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y. Zeevi. The farthest point strategy for progressive image sampling. *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 2 - Conference B: Computer Vision & Image Processing. (Cat. No.94CH3440-5)*, pages 93–97 vol.3, 1994.

Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid DFT error. *Journal of Chemical Theory and Computation*, 13(11):5255–5264, oct 2017. doi: 10.1021/acs.jctc.7b00577.

Dan Feldman. Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 23–44. Springer International Publishing, oct 2019. doi: 10.1007/978-3-030-29349-9_2.

A. M. Frieze. A cost function property for plant location problems. *Mathematical Programming*, 7(1):245–248, dec 1974. doi: 10.1007/bf01585521.

Satoru Fujishige. *Submodular Functions and Optimization, Volume 58.* Elsevier Science, 2005.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, dec 2020. doi: 10.1007/s10994-020-05929-w.

Tim Gould and Stephen G. Dale. Poisoning density functional theory with benchmark sets of difficult systems. *Phys. Chem. Chem. Phys.*, 24:6398–6403, 2022. doi: 10.1039/D2CP00268J.

Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, 2007.

Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O. Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The Journal of Physical Chemistry Letters*, 6(12):2326–2331, jun 2015. doi: 10.1021/acs.jpclett.5b00831.

Sariel Har-Peled. *Geometric approximation algorithms.* American Mathematical Society, 2011.

Dorit S. Hochbaum. When are NP-hard location problems easy? *Annals of Operations Research*, 1(3):201–214, oct 1984. doi: 10.1007/bf01874389.

Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, may 1985. doi: 10.1287/moor.10.2.180.

R. C. St. John and N. R. Draper. D-optimality for regression designs: A review. *Technometrics*, 17(1):15–23, feb 1975. doi: 10.1080/00401706.1975.10489266.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3: 71–104, 2014.

G. Landrum. Rdkit:. *Open-source cheminformatics*, 2012. URL `http://www.rdkit.org`.

Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, jan 2009. doi: 10.1073/pnas.0803205106.

Shie Mannor, Xin Jin, Jiawei Han, Xin Jin, Jiawei Han, Xin Jin, Jiawei Han, and Xinhua Zhang. K-medoids clustering. In *Encyclopedia of Machine Learning*, pages 564–565. Springer US, 2011. doi: 10.1007/978-0-387-30164-8_426.

Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Douwe Kiela, David Jurado, David Kanter, Rafael Mosquera, Juan Ciro, Lora Aroyo, Bilge Acun, Sabri Eyuboglu, Amirata Ghorbani, Emmett Goodman, Tariq Kane, Christine R. Kirkpatrick, Tzu-Sheng Kuo, Jonas Mueller, Tristan Thrush, Joaquin Vanschoren, Margaret Warren, Adina Williams, Serena Yeung, Newsha Ardalani, Praveen Paritosh, Ce Zhang, James Zou, Carole-Jean Wu, Cody Coleman, Andrew Ng, Peter Mattson, and Vijay Janapa Reddi. Dataperf: Benchmarks for data-centric ai development, 2022.

Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, sep 2013. doi: 10.1088/1367-2630/15/9/095003.

Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, 10(1), feb 2018. doi: 10.1186/s13321-018-0258-y.

K. Murphy. *Probabilistic Machine Learning: An Introduction.* MIT Press, Cambridge, MA, USA, 2022. ISBN 9780262182539.

Gabriel A. Pinheiro, Johnatan Mucelini, Marinalva D. Soares, Ronaldo C. Prati, Juarez L. F. Da Silva, and Marcos G. Quiles. Machine learning prediction of nine molecular properties based on the SMILES representation of the QM9 quantum-chemistry dataset. *The Journal of Physical Chemistry A*, 124(47):9854–9866, nov 2020. doi: 10.1021/acs.jpca.0c05969.

Wiktor Pronobis, Kristof T. Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Capturing intensive and extensive DFT/TDDFT molecular properties with machine learning. *The European Physical Journal B*, 91(8), aug 2018. doi: 10.1140/epjb/e2018-90148-y.

Raghunathan Ramakrishnan and O. Anatole von Lilienfeld. Machine learning, quantum chemistry, and chemical space. In *Reviews in Computational Chemistry*, pages 225–256. John Wiley & Sons, Inc., apr 2017. doi: 10.1002/9781119356059.ch5.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O. Anatole von Lilienfeld. Electronic spectra from TDDFT and machine learning in chemical space. *The Journal of Chemical Physics*, 143(8), aug 2015. doi: 10.1063/1.4928757.

Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, nov 2012. doi: 10.1021/ci300415d.

M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108:058301, 2012.

Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: Analysis and efficient estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3839–3848, Red Hook, NY, USA, 2018. Curran Associates Inc.

Patrick Schober, Christa Boer, and Lothar A. Schwarte. Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768, may 2018. doi: 10.1213/ane.0000000000002864.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1aIuk-RW`.

B. Settles. *Active Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

Dravyansh Sharma, Ashish Kapoor, and Amit Deshpande. On greedy maximization of entropy. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1330–1338, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/sharma15.html`.

Annika Stuke, Milica Todorović, Matthias Rupp, Christian Kunkel, Kunal Ghosh, Lauri Himanen, and Patrick Rinke. Chemical diversity in molecular orbital energy predictions with kernel ridge regression. *The Journal of Chemical Physics*, 150(20):204121, may 2019. doi: 10.1063/1.5086105.

Christopher Sutton, Mario Boley, Luca M. Ghiringhelli, Matthias Rupp, Jilles Vreeken, and Matthias Scheffler. Identifying domains of applicability of machine learning models for materials science. *Nature Communications*, 11(1):4428, sep 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-17112-9.

Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(13):363–392, 2005.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1954–1963, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/wei15.html`.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling*, 28 (1):31–36, feb 1988. doi: 10.1021/ci00057a005.

Holger Wendland. *Scattered Data Approximation.* Cambridge University Press, dec 2004. doi: 10.1017/cbo9780511617539.

Dongrui Wu, Chin-Teng Lin, and Jian Huang. Active learning for regression using greedy sampling. *Information Sciences*, 474:90–105, feb 2019. doi: 10.1016/j.ins.2018.09.060.

Hwanjo Yu and Sungchul Kim. Passive sampling for regression. In *2010 IEEE International Conference on Data Mining.* IEEE, dec 2010. doi: 10.1109/icdm.2010.9.

Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning - ICML '06.* ACM Press, 2006. doi: 10.1145/1143844.1143980.

Viktor Zaverkin, David Holzmüller, Ingo Steinwart, and Johannes Kästner. Exploring chemical and conformational spaces by batch mode deep active learning. *Digital Discovery*, 2022. doi: 10.1039/D2DD00034B.