Investigating minimizing the training set fill distance in machine learning regression

Paolo Climaco [1] **Jochen Garcke** [1 2]

Abstract

Many machine learning regression methods leverage large datasets for training predictive models. However, using large datasets may not be feasible due to computational limitations or high labelling costs. Therefore, sampling small training sets from large pools of unlabelled data points is essential to maximize model performance while maintaining computational efficiency. In this work, we study a sampling approach aimed to minimize the fill distance of the selected set. We derive an upper bound for the maximum expected prediction error that linearly depends on the training set fill distance, conditional to the knowledge of data features. For empirical validation, we perform experiments using two regression models on two datasets. We empirically show that selecting a training set by aiming to minimize the fill distance, thereby minimizing the bound, significantly reduces the maximum prediction error of various regression models, outperforming existing sampling approaches by a large margin.

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) and force field approximation (Chmiela et al., 2017; Unke et al., 2021). 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 an efficient training set is crucial for the quality of the model predictions. In this context, our focus on selecting data points that can improve the quality of predictions for different 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, 2009) involves iteratively updating the parameters of one or several regression models and predicting uncertainties for unlabelled samples. Unfortunately, it typically only benefits a specific model or model class and optimizes the models' performance for a specific learning task, as it exploits the labels' knowledge to iteratively update the models' parameters during the selection process. Passive sampling (Yu & 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 believe 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 & 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 selected set's fill distance, that is, the maximum distance between a point in the set and its nearest selected element.

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

[1]Institut für Numerische Simulation, Universität Bonn, 53115 Bonn, Germany. [2]Fraunhofer SCAI, 53754 Sankt Augustin, Germany. Correspondence to: Paolo Climaco <climaco@ins.uni-bonn.de>.

shown that minimizing training set fill distance reduces the average prediction error of Lipschitz continuous classification models with soft-max output layer and bounded error function (Sener & 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 & 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 can be considered as a measure of the robustness of a model's 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 a model's predictions (Sutton et al., 2020; Gould & Dale, 2022). Our analysis offers theoretical and empirical results, which set it apart from previous works. Specifically, we extend the theoretical work done in (Sener & Savarese, 2018) for classification to regression, demonstrating that reducing training set fill distance lowers the regression model's maximum prediction error. Moreover, contrary to (Yu & 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 the benefits we highlight regarding reducing the training set fill distance using FPS were not detected in previous works, either theoretically or empirically.

2. Related work

Existing work concerning model-agnostic passive sampling is mostly related to coresets approaches. Coresets (Feldman, 2019) identify the most informative training data subset. 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 & Drineas, 2009), assign to samples relevance-based weights. Grid-based methods, such as k-medoids and k-medoids++ (Mannor et al., 2011), that are adapted version of the k-means (Ahmed et al., 2020) and k-means++ (Arthur & Vassilvitskii, 2007), 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 & 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 & Schuurmans, 2007), linear regression with Gaussian noise (Yu et al., 2006) and support vector machines (Tsang et al., 2005). Assuming the learning model's knowledge may even lead to the development of optimal training set selection strategies, as in the case of linear regression (Yu et al., 2006). Unfortunately, these selection strategies benefit only specific model classes. 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 selected set's fill distance 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 & Savarese, 2018). In (Sener & Savarese, 2018) the authors show that selecting the training set by fill distance minimization can reduce the average prediction 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 & 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 model's prediction for the given data point. The smaller the error, the better the prediction.

Furthermore, we consider a dataset $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^k \subset \mathcal{X} \times \mathcal{Y}$, $k \in \mathbb{N}$, 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 $\{\boldsymbol{x}_i\}_{i=1}^k$, while the labels $\{y_i\}_{i=1}^k$ 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 $\mathcal{L} := \{(\boldsymbol{x}_{i_j}, y_{i_j})\}_{j=1}^b \subset \mathcal{D}$, $b \ll k$, with $i_j \in \{1, 2, \dots, k\}$ $\forall j$. 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. The symbol $2^{\mathcal{D}}$ represents the set of all possible subsets of $\mathcal{D}$. In what follows, unless otherwise specified, the labelled points in the selected set $\mathcal{L}$ are indexed with j, that is, $\mathcal{L} := \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b$, while the unlabelled points are indexed with i, that is, $\mathcal{U} := \mathcal{D} - \mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $n = k - b$ and the labels $\{y_i\}_{i=1}^n$ are unknown. Furthermore, given a set $\mathcal{L} := \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b \subset \mathcal{D}$ we define $\mathcal{L}_{\mathcal{X}} := \{\boldsymbol{x}_j\}_{j=1}^b$, $\mathcal{L}_{\mathcal{Y}} := \{y_j\}_{j=1}^b$.

In several applications the labelling process is computationally expensive, therefore, given a budget $b \ll n$ 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 labels' predictions obtained with the learned function. Specifically, the problem we want to solve can be expressed as follows:

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

where $p_{Y|\boldsymbol{X}}$ is the conditional probability of the random variable Y given $\boldsymbol{X} = \boldsymbol{x}$, that will be formally introduced later. 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 the trained regression model $m_{\mathcal{L}}$ evaluated on the unlabelled points is minimized. We remark that this work focuses 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 a specific learning model class.

4. Method

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) depending linearly on the training set fill distance, a quantity that can be optimized. 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, we introduce the concept of fill distance, a quantity we can associate with subsets of the pool of data points we wish to label that can be calculated only considering the data points' features.

Definition 4.1. Given $\mathcal{U}_{\mathcal{X}} := \{\boldsymbol{x}_i\}_{i=1}^n$ and $\mathcal{L}_{\mathcal{X}} = \{\boldsymbol{x}_j\}_{j=1}^b$ disjoint subsets of $\mathcal{X} \subset \mathbb{R}^d$, the fill distance of $\mathcal{L}_{\mathcal{X}}$ in $\mathcal{U}_{\mathcal{X}}$ is defined as

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

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

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, for both the feature and label spaces, but the following result can be generalized to other distance metrics.

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

Assumption 4.2. We assume there exists $\epsilon \geq 0$ such that

for each data point $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ we have that

$$p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) = 0 \text{ almost everywhere outside } [y_i - \epsilon, y_i + \epsilon] \tag{3}$$

where

$$p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) := \frac{p_{\mathcal{Z}}(\boldsymbol{x}_i, y)}{p_{\boldsymbol{X}}(\boldsymbol{x}_i)} \text{ and } p_{\boldsymbol{X}}(\boldsymbol{x}_i) := \int_{\mathcal{Y}} p_{\mathcal{Z}}(\boldsymbol{x}_i, y)dy.$$

We refer to 'ϵ' as the labels' uncertainty. Moreover, we assume $p_{Y|\boldsymbol{X}}$ to be λ^p-Lipschitz continuous, that is,

$$|p_{Y|\boldsymbol{X}}(y|\hat{\boldsymbol{x}}) - p_{Y|\boldsymbol{X}}(y|\tilde{\boldsymbol{x}})| \leq \lambda^p \|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}\|_2, \tag{4}$$

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

Thus, from (3) we have that given a realization $\boldsymbol{X} = \boldsymbol{x}_i$, with probability 1 the realizations of the random variable Y take value in an ϵ-neighborhood of a specific point $y_i \in \mathcal{Y}$, that is, given $\boldsymbol{x}_i \in \mathcal{X}$, a point in the feature space, the value of the associated label is not fixed but localized in a "small" region of the label space. This assumption on the data aims to model those scenarios where the underlying true mapping between the feature and label spaces is either stochastic in nature or deterministic but subject to random fluctuations with a maximum magnitude of ϵ. The Lipschitz continuity in (4) 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 probabilities 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 pertains to the error function used to evaluate the model's performance and the model's prediction quality 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 confine our analysis to error functions that exhibit a certain level of regularity.

Assumption 4.3. We assume there exist $\epsilon_{\mathcal{L}} \geq 0$, depending on the labelled set $\mathcal{L} \subset \mathcal{D}$, such that for each labelled point $(\boldsymbol{x}_j, y_j) \in \mathcal{L}$ we have that

$$l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) \leq \epsilon_{\mathcal{L}} \; \forall \; y \in [y_j - \epsilon, y_j + \epsilon], \tag{5}$$

where ϵ is the label's uncertainty introduced in Assumption 4.2. We consider $\epsilon_{\mathcal{L}}$ as the maximum 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.

With (5) we assume that the error on the training set is bounded, and the bound takes into account that the realizations of the variable Y given $\boldsymbol{X} = \boldsymbol{x}_j$ may be affected by small variations. 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 A.1, at the end of Appendix A. For instance, the $\lambda^{l_{\mathcal{Y}}}$-Lipschitz regularity is verified by all L_q-norm error functions, with $1 \leq q < \infty$.

Finally, we introduce 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 4.4. *Given $\mathcal{U} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and $\mathcal{L} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b$ disjoint sets 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 4.2 and 4.3 are fulfilled, then we have that*

$$\max_{(\boldsymbol{x},y)\in\mathcal{U}} \mathbb{E}_{p_{Y|\boldsymbol{X}}}[l(\boldsymbol{x}, y, m_{\mathcal{L}})|\boldsymbol{x}] \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}}\left(\lambda^{l_{\mathcal{X}}} + \mathcal{O}(\epsilon)\right) + \epsilon_{\mathcal{L}}, \tag{6}$$

where $h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}}$ is the fill distance of $\mathcal{L}_{\mathcal{X}}$ in $\mathcal{U}_{\mathcal{X}}$, ϵ is the labels' uncertainty from assumption 4.2, $\lambda^{l_{\mathcal{X}}}$ is the Lipschitz constant of the error function, and $\epsilon_{\mathcal{L}}$ is the maximum error of the trained model's predictions on the labelled set $\mathcal{L}$.

The proof can be found in Appendix A. Formula (6) provides an upper bound for the minimization objective in (1) that is linearly dependent on the fill distance. Therefore, 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 knowledge of the data features. 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. Note that the bound shown also depends on the labels' uncertainty 'ϵ'. In particular, the larger the label uncertainty, the larger the bound for a fixed training set fill distance. Additionally, the connection between the bound and the regularity of the chosen error function is highlighted by the presence of the Lipschitz constant $\lambda^{l_{\mathcal{X}}}$ of the error function on the right-hand side of (6). If we consider the error function to be the L_2-distance between true and predicted labels, Theorem 4.4 holds for all Lipschitz continuous regression models, such as kernel ridge regression with Gaussian kernel and feed forward neural networks, as we explain in Remark A.1 in Appendix A.

4.2. Selecting training sets with farthest point sampling

Theorem 4.4 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 error on the training set is negligible, we can attempt the minimization of the upper bound in (6) by solving the following minimization problem

$$\min_{\substack{\mathcal{L}\subset\mathcal{D},\\ |\mathcal{L}|=b}} h_{\mathcal{L}_\mathcal{X},\mathcal{U}_\mathcal{X}} \tag{7}$$

where $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^k \subset \mathcal{X} \times \mathcal{Y}$ is the pool of data points we want to label, $\mathcal{L} := \{(\boldsymbol{x}_j, y_j)\}_{j=1}^b$ is the set of labelled points we use for training and $\mathcal{U} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is the set of unlabelled point for which we want to compute the predictions, with $k = n + b$. Notice that $\mathcal{D} = \mathcal{L} \sqcup \mathcal{U}$, thus $h_{\mathcal{L}_\mathcal{X},\mathcal{U}_\mathcal{X}} = h_{\mathcal{L}_\mathcal{X},\mathcal{D}_\mathcal{X}}$. The minimization problem in (7) 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, thus, so that 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 training sets with fill distance at most a factor of 2 from the minimal fill distance (Har-Peled, 2011). Assume $O \subset \mathcal{D}$ is a subset of cardinality b with minimal fill distance, that is,

$$O := \arg\min_{\substack{\mathcal{L}\subset\mathcal{D},\\ |\mathcal{L}|=b}} h_{\mathcal{L}_\mathcal{X},\mathcal{U}_\mathcal{X}}. \tag{8}$$

Then, the fill distance of a set $\mathcal{L}^{FPS} \subset \mathcal{D}$, $|\mathcal{L}^{FPS}| = b$, obtained using FPS, is at most two times the minimal fill distance, that is,

$$h_{\mathcal{L}_\mathcal{X}^{FPS},\mathcal{D}_\mathcal{X}} \leq 2h_{O_\mathcal{X},\mathcal{D}_\mathcal{X}}. \tag{9}$$

FPS can be implemented using $\mathcal{O}(|\mathcal{D}|)$ space and takes $\mathcal{O}(|\mathcal{D}||\mathcal{L}^{FPS}|)$ 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 & Shmoys, 1985). Thus, FPS provides a suboptimal solution, but obtaining a better approximation with theoretical guarantees would not be feasible in polynomial time. According to our recent experiments, it takes approximately 17 minutes 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 $\mathbb{R}^{256}$. We used the Deep Graph python library (Wang et al., 2019) to implement FPS on a 48-cores CPU with 384 GB RAM. Such experiments give a qualitative understanding of the data efficiency of FPS.

Algorithm 1 Farthest Point Sampling (FPS)

Input Dataset $\mathcal{D}_\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^k \subset \mathcal{X}$ 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**
  $\bar{\boldsymbol{x}} = \arg\max_{x_i \in \mathcal{D}_\mathcal{X}} \min_{x_j \in \mathcal{L}_\mathcal{X}^{FPS}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$.
  $\mathcal{L}_\mathcal{X}^{FPS} \leftarrow \mathcal{L}_\mathcal{X}^{FPS} \cup \bar{\boldsymbol{x}}$.
end while

5. Experimental results

5.1. Datasets

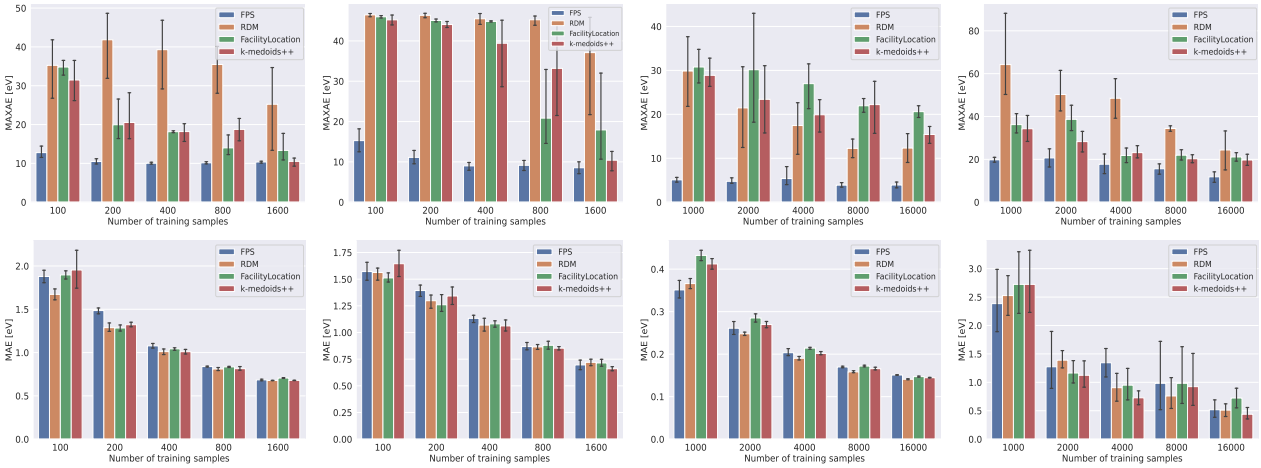
We employ the datasets QM7 and QM9 in our experiments, where the task is to predict the atomization energy. Additional information on the datasets, preprocessing procedures and used descriptors are provided in Appendix B.

QM7 (Blum & Reymond, 2009; Rupp et al., 2012) is a benchmark dataset in quantum chemistry, consisting of 7165 organic molecules with up to 23 atoms. It includes information, such as the atoms Cartesian coordinates, and the molecules' atomization energy. We use the QM7 for a regression task, where each molecule's feature vector is the Coulomb matrix (Rupp et al., 2012), that in the case of the QM7 can be represented as an element in $\mathbb{R}^{529}$, and the label value to predict is the atomization energy, measured in electronvolt (eV).

QM9 (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014) is a publicly available quantum chemistry dataset containing the properties of 133,885 organic molecules with up to nine heavy atoms. The 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 & von Lilienfeld, 2017; Pronobis et al., 2018). Each molecule in our dataset is represented by a vector in $\mathbb{R}^{1307}$, describing the molecule's topological structure, and the label value to predict is the atomization energy measured in eV.

5.2. 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 the kernel ridge regression with Gaussian kernel (KRR) (Stuke et al., 2019; Deringer et al., 2021) and the 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 A.1, we know is a required property to validate our theoretical analysis. A detailed description of the learning models used in this work and information related to their



(a) KRR on QM7 (b) FNN on QM7 (c) KRR on QM9 (d) FNN on QM9

Figure 1. Results for atomization energy regression task on QM7 and QM9 using KRR and FNN trained on sets of various sizes and selected with different sampling strategies. MAXAE (top row) and MAE (bottom row) of the predictions are shown for each regression model, training set size and sampling approach.

Lipschitz continuity are provided in Appendix C.

5.3. 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,

$$\text{MAXAE} := \max_{1\leq i\leq n} |y_i - \tilde{y}_i|, \tag{10}$$

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,

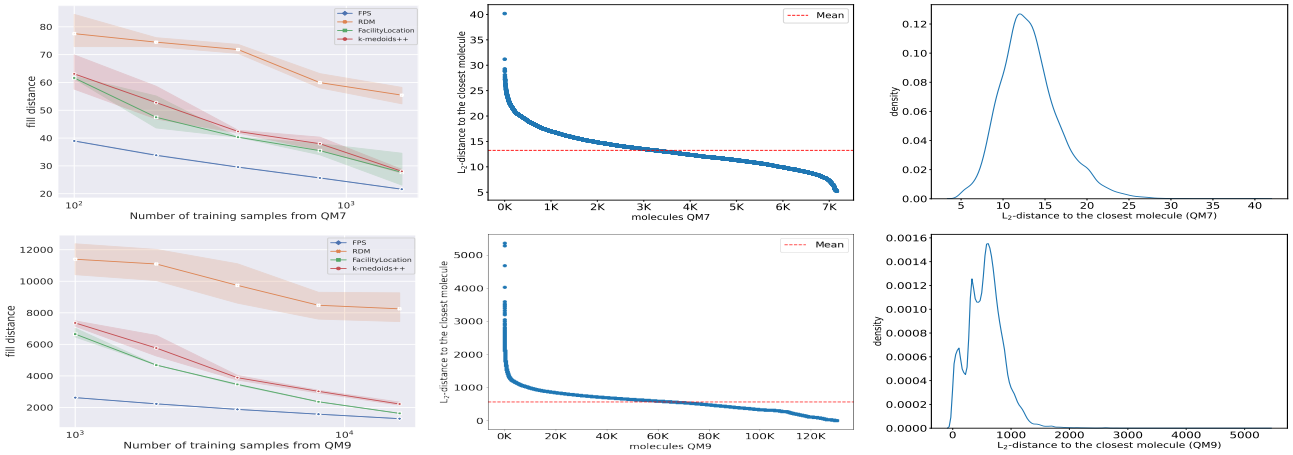
$$\text{MAE} := \frac{1}{n}\sum_{i=1}^{n} |y_i - \tilde{y}_i|. \tag{11}$$

5.4. Numerical Results

We investigate the effects of minimizing the training set fill distance for the atomization energy regression task on the QM7 and QM9 datasets using both KRR and FNN.

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 & 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.

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 predictions' 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 bars representing the results' standard deviation. We remark that, our experiments' final goal 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.



(a) Training sets fill distances. (b) Distances to nearest neighbour. (c) Density distances to nearest neighbour.

Figure 2. (a) Selected training sets fill distances. (b) Euclidean distances to the nearest neighbour and (c) density of such distances for molecules in the QM7 (top 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.

5.4.1. Molecular energy prediction on QM7 and QM9 datasets

Fig. 1 shows the results for the atomization energy regression task on the QM7 and QM9 datasets using the KRR and FNN as learning models. The graphs on the top row of Fig. 1, illustrating the maximum error of the predictions on the unlabelled points, 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 show the MAE of the predictions on the QM7 and QM9 datasets for KRR and FNN. The graphs indicate that selecting training sets with FPS doesn't drastically reduce the predictions' MAE on the unlabelled points with respect to the baselines, independently of the dataset and regression model. On the contrary, we can provide examples where FPS performs worse than the baselines, e.g., with the FNN on the QM7 and QM9 for training set sizes of 400 and 4000, respectively. These experiments suggest that, contrary to what has been shown for classification (Sener & 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.

Notice that KRR outperforms the FNN. This is attributed to the fact that Neural Networks are generally less data-efficient than KRR models and require more data points for effective training of their parameters, but they can scale better to larger datasets (Schütt et al., 2017).

5.4.2. Empirical analysis and discussion

Interestingly, with FPS, the MAXAE converges fast to a plateau value for both datasets and regression models (Fig. 1). 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 predictions' MAXAE are directly correlated with the fill distances of the respective labelled sets used for training, illustrated in Fig. 2a. From Fig. 2a 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 predictions MAXAE. Nevertheless, our theoretical analysis shows that the training set fill distance is only linked to the maximum 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 on the unlabelled set 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 predictions MAXAE, but as a qualitative indicator of the model robustness that, if minimized, leads to a substantial reduction of the predictions MAXAE.

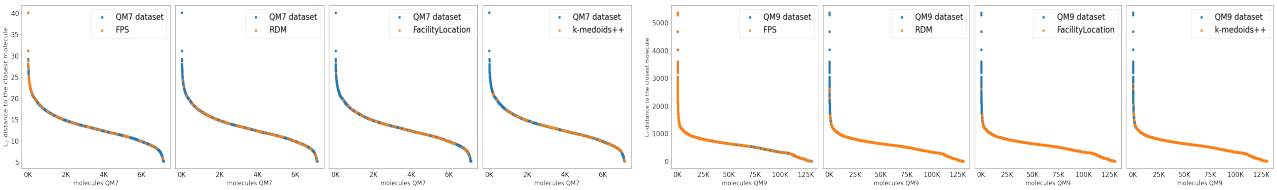


Figure 3. In blue, the Euclidean distances to the nearest neighbour for molecules in the QM7 and QM9. In orange are highlighted the molecules selected with FPS and the other baselines. The sizes of the selected sets are 100 and 1000 for QM7 and QM9, respectively.

Previous work has established that FPS approach provides suboptimal minimizers for the k-center problem (Har-Peled, 2011). Consequently, it can be employed to choose a training set by fill distance minimization since the problem of selecting a training set with minimal fill distance is essentially a k-center problem, as we mentioned in Subsection 4.2. This insight, together with our proposed bound for the maximum expected prediction error, linearly dependent on the fill distance, provides a theoretical motivation for the effectiveness of FPS in reducing the predictions' MAXAE. We now aim to provide a more empirical motivation for 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. 2b and Fig. 2c, showing for each molecule the Euclidean distance to the respective closest molecule and the density of such distances, respectively, for the QM7 and QM9 datasets. Fig. 2b shows that, in both 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. 2c, representing the density distribution of the molecules' distances to the 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 molecules even for a low training data budget. Fig. 3 highlights the Euclidean distances to the closest neighbour for molecules selected with FPS, and the other baseline strategies, from the QM7 and QM9 datasets. FPS, facility location, and k-medoids++ have been initialized with the same random element for better comparison. The size of the selected sets is 100 and 1000 for the QM7 and QM9, respectively. Specifically, we are analyzing the same elements selected in the lowest training data budget we considered for the atomization energy regression tasks in Fig. 1. Fig. 3 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. 2c).

Our hypothesis that selecting isolated molecules is beneficial in terms of the MAXAE reduction is also supported by our theoretical analysis. As a matter of fact, from Theorem 4.4, we know that the maximum expected error of the predictions on the unlabelled dataset is directly correlated with the fill distance, that is, the maximal distance in the feature space between the points for which we want to predict the labels and their closest selected element. Consequently, a sampling strategy that aims to reduce the prediction's maximum error should include the isolated molecules in the training set, as their distance to the nearest neighbour is much larger than the average.

6. Conclusion

We study the effects of minimizing the training set fill distance for Lipschitz continuous regression models. Our numerical results have shown that using FPS to select training sets by fill distance minimization increases the models robustness by significantly reducing the prediction maximum error, in correspondence to our theoretical motivation.

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 that there may be more convenient sampling strategies than FPS to select points at the center of the distribution, where more information is available. Further research in this direction is warranted.

References

Ahmed, M., Seraj, R., and Islam, S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, aug 2020. doi: 10.3390/electronics9081295.

Arthur, D. and Vassilvitskii, S. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

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

Cersonsky, R. K., Helfrecht, B. A., Engel, E. A., Kliavinek, S., and Ceriotti, M. 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.

Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., and Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5), may 2017. doi: 10.1126/sciadv.1603015.

Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., and Csányi, G. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121 (16):10073–10141, aug 2021. doi: 10.1021/acs.chemrev.1c00022.

Eldar, Y., Lindenbaum, M., Porat, M., and Zeevi, Y. Y. 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)*, pp. 93–97 vol.3, 1994.

Faber, F. A., Hutchison, L., Huang, B., Gilmer, J., Schoenholz, S. S., Dahl, G. E., Vinyals, O., Kearnes, S., Riley, P. F., and von Lilienfeld, O. A. 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.

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

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

Fujishige, S. *Submodular Functions and Optimization, Volume 58*. Elsevier Science, 2005. ISBN 9780444520869.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

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

Gould, T. and Dale, S. G. Poisoning density functional theory with benchmark sets of difficult systems. *Phys. Chem. Chem. Phys.*, 24:6398–6403, 2022. doi: 10.1039/D2CP00268J. URL http://dx.doi.org/10.1039/D2CP00268J.

Guo, Y. and Schuurmans, D. Discriminative batch mode active learning. In Platt, J., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/ccc0aa1b81bf81e16c676ddb977c5881-Paper.pdf.

Hansen, K., Biegler, F., Ramakrishnan, R., Pronobis, W., von Lilienfeld, O. A., Müller, K.-R., and Tkatchenko, A. 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.

Har-Peled, S. *Geometric approximation algorithms*. American Mathematical Society, 2011. ISBN 9780821849118.

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

Hochbaum, D. S. and Shmoys, D. B. 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.

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

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

Mahoney, M. W. and Drineas, P. 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.

Mannor, S., Jin, X., Han, J., Jin, X., Han, J., Jin, X., Han, J., and Zhang, X. K-medoids clustering. In *Encyclopedia of Machine Learning*, pp. 564–565. Springer US, 2011. doi: 10.1007/978-0-387-30164-8_426.

Mazumder, M., Banbury, C., Yao, X., Karlaš, B., Rojas, W. G., Diamos, S., Diamos, G., He, L., Kiela, D., Jurado, D., Kanter, D., Mosquera, R., Ciro, J., Aroyo, L., Acun, B., Eyuboglu, S., Ghorbani, A., Goodman, E., Kane, T., Kirkpatrick, C. R., Kuo, T.-S., Mueller, J., Thrush, T., Vanschoren, J., Warren, M., Williams, A., Yeung, S., Ardalani, N., Paritosh, P., Zhang, C., Zou, J., Wu, C.-J., Coleman, C., Ng, A., Mattson, P., and Reddi, V. J.

Dataperf: Benchmarks for data-centric ai development, 2022.

Montavon, G., Rupp, M., Gobre, V., Vazquez-Mayagoitia, A., Hansen, K., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. 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.

Moriwaki, H., Tian, Y.-S., Kawashita, N., and Takagi, T. Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, 10(1), feb 2018. doi: 10.1186/s13321-018-0258-y.

Pinheiro, G. A., Mucelini, J., Soares, M. D., Prati, R. C., Silva, J. L. F. D., and Quiles, M. G. 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.

Pronobis, W., Schütt, K. T., Tkatchenko, A., and Müller, K.-R. 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.

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

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Ruddigkeit, L., van Deursen, R., Blum, L. C., and Reymond, J.-L. 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.

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

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

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Sener, O. and Savarese, S. 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.

Settles, B. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Sharma, D., Kapoor, A., and Deshpande, A. On greedy maximization of entropy. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1330–1338, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sharma15.html.

Stuke, A., Todorović, M., Rupp, M., Kunkel, C., Ghosh, K., Himanen, L., and Rinke, P. 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.

Sutton, C., Boley, M., Ghiringhelli, L. M., Rupp, M., Vreeken, J., and Scheffler, M. 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. URL https://www.nature.com/articles/s41467-020-17112-9.

Tsang, I. W., Kwok, J. T., and Cheung, P.-M. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(13):363–392, 2005. URL http://jmlr.org/papers/v6/tsang05a.html.

Unke, O. T., Chmiela, S., Sauceda, H. E., Gastegger, M., Poltavsky, I., Schütt, K. T., Tkatchenko, A., and Müller, K.-R. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, mar 2021. doi: 10.1021/acs.chemrev.0c01111.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis,

G., Li, J., and Zhang, Z. Deep graph library: A graph-centric, highly-performant package for graph neural networks, 2019.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1954–1963, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wei15.html.

Weininger, D. 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.

Wu, D., Lin, C.-T., and Huang, J. Active learning for regression using greedy sampling. *Information Sciences*, 474: 90–105, feb 2019. doi: 10.1016/j.ins.2018.09.060.

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

Yu, K., Bi, J., and Tresp, V. 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.

Zaverkin, V., Holzmüller, D., Steinwart, I., and Kästner, J. Exploring chemical and conformational spaces by batch mode deep active learning. *Digital Discovery*, 2022. doi: 10.1039/D2DD00034B. URL http://dx.doi.org/10.1039/D2DD00034B.

A. Proof Theorem 4.4

Proof. First we want to find an upper bound for $\mathbb{E}_{p_{Y|\boldsymbol{X}}}[l(\boldsymbol{x}_i, Y, m_{\mathcal{L}})|\boldsymbol{x}_i]$ for each $\boldsymbol{x}_i \in \mathcal{U}_{\mathcal{X}},\ i = 1, \dots, n$. Fixed $\boldsymbol{x}_i \in \mathcal{U}_{\mathcal{X}}$, by definition of fill distance we know there exists $\boldsymbol{x}_j \in \mathcal{L}_{\mathcal{X}}$ such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}}$.

$$\begin{aligned}
\mathbb{E}_{p_{Y|\boldsymbol{X}}}[l(\boldsymbol{x}_i, Y, m_{\mathcal{L}})|\boldsymbol{x}_i] &= \int_{\mathcal{Y}} l(\boldsymbol{x}_i, y, m_{\mathcal{L}}) p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) dy \\
&\leq \int_{\mathcal{Y}} |l(\boldsymbol{x}_i, y, m_{\mathcal{L}}) - l(\boldsymbol{x}_j, y, m_{\mathcal{L}})| p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) dy + \int_{\mathcal{Y}} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) dy \qquad (12) \\
&\leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \lambda^{l_{\mathcal{X}}} + \int_{\mathcal{Y}} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) dy
\end{aligned}$$

where $\lambda^{l_{\mathcal{X}}}$ from Assumption 4.3 and $\boldsymbol{x}_j \in \mathcal{L}_{\mathcal{X}}$ such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}}$. The second inequality in (12) follows from the $\lambda^{l_{\mathcal{X}}}$-Lipschitz continuity of the error function. We can bound the remaining term as follows

$$\begin{aligned}
\int_{\mathcal{Y}} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) dy &\leq \int_{y_i-\epsilon}^{y_i+\epsilon} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) |p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_i) - p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_j)| dy + \int_{y_i-\epsilon}^{y_i+\epsilon} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}_j) dy \\
&\leq \lambda^p h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \int_{y_i-\epsilon}^{y_i+\epsilon} l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) dy + \epsilon_{\mathcal{L}} \\
&\leq \lambda^p h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \left(\int_{y_i-\epsilon}^{y_i+\epsilon} |l(\boldsymbol{x}_j, y, m_{\mathcal{L}}) - l(\boldsymbol{x}_j, y_j, m_{\mathcal{L}})| dy + \int_{y_i-\epsilon}^{y_i+\epsilon} l(\boldsymbol{x}_j, y_j, m_{\mathcal{L}}) dy \right) + \epsilon_{\mathcal{L}} \\
&\leq \lambda^p h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \left(\lambda^{l_{\mathcal{Y}}} \int_{y_i-\epsilon}^{y_i+\epsilon} (\|y - y_i\|_2 + \|y_i - y_j\|_2) dy + 2\epsilon\epsilon_{\mathcal{L}} \right) + \epsilon_{\mathcal{L}} \\
&\leq \lambda^p h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \left(2\epsilon\lambda^{l_{\mathcal{Y}}} \left(\epsilon + \max_{\substack{y_i \in \mathcal{U}_{\mathcal{Y}} \\ y_j \in \mathcal{L}_{\mathcal{Y}}}} \|y_i - y_j\|_2 \right) + 2\epsilon\epsilon_{\mathcal{L}} \right) + \epsilon_{\mathcal{L}} \\
&= h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \epsilon\lambda^p M + \epsilon_{\mathcal{L}}
\end{aligned} \qquad (13)$$

where $M := 2\left(\lambda^{l_{\mathcal{Y}}} \left(\epsilon + \max_{\substack{y_i \in \mathcal{U}_{\mathcal{Y}} \\ y_j \in \mathcal{L}_{\mathcal{Y}}}} \|y_i - y_j\|_2 \right) + \epsilon_{\mathcal{L}} \right)$. The integration's range change after the first inequality in (13) follows from Assumption 4.2. The second inequality follows from the λ^p-Lipschitz continuity of the conditional probability $p_{Y|\boldsymbol{X}}$ and the fourth inequality from the $\lambda^{l_{\mathcal{Y}}}$-Lipschitz continuity of the error function. Since the above inequality holds for each $\boldsymbol{x}_i \in \mathcal{U}_x$, we have that

$$\max_{1 \leq i \leq n} \mathbb{E}_{p_{Y|\boldsymbol{X}}}[l(\boldsymbol{x}_i, y_i, m_{\mathcal{L}})|\boldsymbol{x}_i] \leq h_{\mathcal{L}_{\mathcal{X}},\mathcal{U}_{\mathcal{X}}} \left(\lambda^{l_{\mathcal{X}}} + \mathcal{O}(\epsilon) \right) + \epsilon_{\mathcal{L}}, \qquad (14)$$

□

Remark A.1. If the trained model $m_{\mathcal{L}} \in \mathcal{M}$ is $\lambda^{l_{\mathcal{X}}}$−Lipschitz continuous, then also the L_2-norm 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

$$|l(\boldsymbol{x}, y, m_{\mathcal{L}}) - l(\tilde{\boldsymbol{x}}, y, m_{\mathcal{L}})| = |\|m_{\mathcal{L}}(\boldsymbol{x}) - y\|_2 - \|m_{\mathcal{L}}(\tilde{\boldsymbol{x}}) - y\|_2| \leq \|m_{\mathcal{L}}(\boldsymbol{x}) - m_{\mathcal{L}}(\tilde{\boldsymbol{x}})\|_2.$$

Moreover, the L_2-norm 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

$$|l(\boldsymbol{x}, y, m_{\mathcal{L}}) - l(\boldsymbol{x}, \tilde{y}, m_{\mathcal{L}})| = |\|m_{\mathcal{L}}(\boldsymbol{x}) - y\|_2 - \|m_{\mathcal{L}}(\boldsymbol{x}) - \tilde{y}\|_2| \leq \|y - \tilde{y}\|_2.$$

Finally, it is important to notice that, since the label values are scalars, the L_2-norm error function is the absolute difference between the true and predicted values.

B. Datasets

This appendix provides an extended version of the datasets' description in Subsection 5.1, including additional information related to the datasets, preprocessing procedures and molecular descriptors used.

QM7

The QM7 (Blum & 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 of each atom, and their atomization energy. We use the QM7 for a regression task, where each molecule's feature vector is the Coulomb matrix (Rupp et al., 2012) 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. The Coulomb matrix is defined as

$$\boldsymbol{C}_{i,j} = \begin{cases} \frac{1}{2} z_i^{2.4} & \text{if } i = j \\ \frac{z_i z_j}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2} & \text{if } i \neq j \end{cases} \tag{15}$$

where z_i is the nuclear charge of the i-th atom and $\boldsymbol{r}_i$ is its position.

QM9

The 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). The QM9 is frequently used for developing and testing machine learning models for predicting molecular properties and for exploring the chemical space of small organic molecules (Faber et al., 2017; Ramakrishnan & von Lilienfeld, 2017; Pronobis et al., 2018). The dataset 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 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 519 features for which the values across the dataset have zero variance. Thus, each molecule in our dataset is represented by a vector in $\mathbb{R}^{1307}$. We use the QM9 for the atomization energy regression task using as data features the Mordred-based, vector-valued molecular representation we introduced.

C. Regression models

This appendix provides a detailed description of the regression models used in this work. The Lipschitz continuity of the described models is also discussed.

C.1. Kernel ridge regression with Gaussian kernel (KRR)

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 to transform the input features into a high-dimensional space where the relationship between the input features and output labels is learned. Given a training set $\mathcal{L} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{b}$, the Gaussian kernel is defined as follows:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) := e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}, \tag{16}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{L}_{\mathcal{X}}$ and $\gamma \in \mathbb{R}$ is a kernel hyperparameter to be selected through an optimization process. Given $\boldsymbol{x} \in \mathcal{X}$ its associated predicted label $y(\boldsymbol{x})$ is defined as follows

$$y(\boldsymbol{x}) := \sum_{j=1}^{b} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j). \tag{17}$$

where the vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_b]^T \in \mathbb{R}^b$ is the solution of the following minimization problem

$$\boldsymbol{\alpha} = \arg\min_{\bar{\boldsymbol{\alpha}}} \sum_{j=1}^{b} (y(\boldsymbol{x}_j) - y_j)^2 + \lambda \bar{\boldsymbol{\alpha}}^T \boldsymbol{K} \bar{\boldsymbol{\alpha}}. \tag{18}$$

$\boldsymbol{K}$ is the kernel matrix, i.e., $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and the parameter $\lambda \in \mathbb{R}$ is the so-called regularization parameter that penalizes larger weights. The analytic solution to the minimization problem in (18) is given by

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y} \tag{19}$$

where $\boldsymbol{y} = [y(\boldsymbol{x}_1), y(\boldsymbol{x}_2), \ldots, y(\boldsymbol{x}_b)]^T$. To show the Lipschitz continuity of the KRR with Gaussian kernel, we propose the following lemma:

Lemma C.1. *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 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 $\boldsymbol{\alpha}_{\mathcal{L}} := [\alpha_1, \alpha_2, \ldots, \alpha_b]^T \in \mathbb{R}^b$ 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:

$$y(\boldsymbol{x}) = \sum_{j=1}^{b} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j) = \boldsymbol{\alpha}_{\mathcal{L}}^T \boldsymbol{k}_{\boldsymbol{x}}, \tag{20}$$

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 $e^{-\gamma r^2}$, $r \in \mathbb{R}^+$ □

C.2. Feed Forward Neural Networks (FNNs)

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,

$$y(\boldsymbol{x}) := \phi_l \circ \sigma_l \circ \phi_{l-1} \circ \sigma_{l-1} \circ \cdots \circ \phi_1(\boldsymbol{x}), \tag{21}$$

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

$$\phi_i(\boldsymbol{x}) = \boldsymbol{W}_i(\boldsymbol{x}) + \boldsymbol{b}_i \tag{22}$$

where the weight matrices $\boldsymbol{W}_i$ and the biases $\boldsymbol{b}_i$ are learned by minimizing n L_2-norm 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 already shown in literature (Scaman & Virmaux, 2018; Gouk et al., 2020).