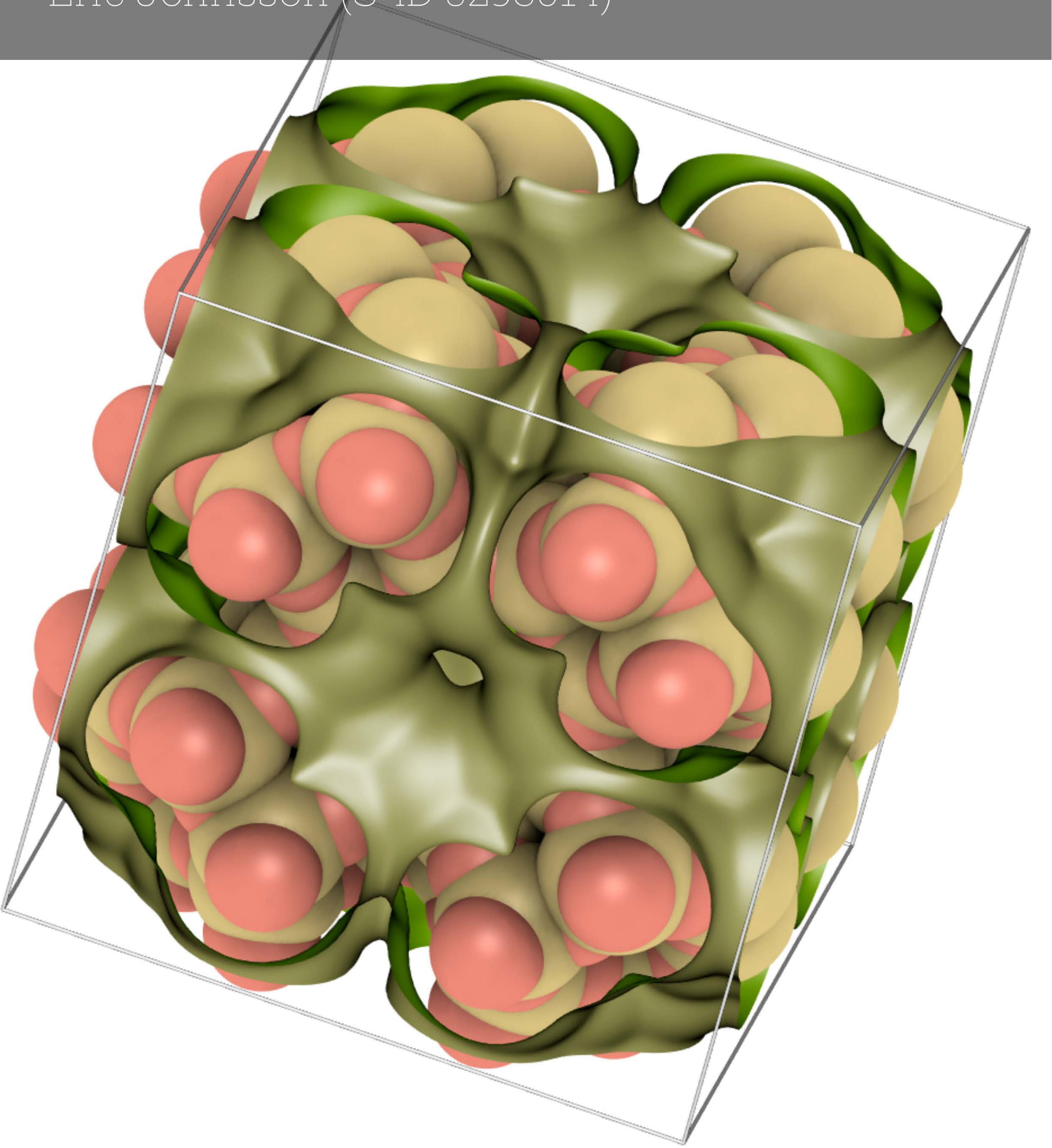
Predicting the Maximum Uptake of Zeolites for Hydroisomerization Applications: A Machine Learning Approach

Eric Johnsson (S-ID 5293014)



Predicting the Maximum Uptake of Zeolites for Hydroisomerization Applications: A Machine Learning Approach

by

Eric Johnsson (S-ID 5293014)

Instructors:	prof. dr. ir. A. Gangoli Rao
	prof. dr. ir. T.J.H. Vlugt
Project Duration:	May, 2025 - January, 2026
Faculty:	Faculty of Aerospace Engineering, Delft
	Faculty of Mechanical Engineering, Delft

Cover: KFI-type zeolite unit cell (iRASPA)

TU Delft

Contents

1

Background Theory

1.1. Adsorption

1.1.1. Physics and Modelling

Adsorption is defined as the enrichment of a compound on an interface or inside a porous structure [1]. For a gas-solid system, this mean molecules of this gas (here referred as adsorbate) clustering on the surface of the solid phase (here referred as the adsorbent) or inside its pores. There are two mechanisms for adsorption: chemisorption and physisorption.

Chemisorption is a type of adsorption that involves the very close chemical bonding between the adsorbate and the adsorbent. This means that energy needs to be supplied for it to happen as it requires crossing an activation barrier. The resulting strong interactions in the adsorbate-adsorbent two mean that the process is often irreversible [2].

Unlike chemisorption, physisorption is reversible. This happens when the adsorbate is in contact with the adsorbing surface. It is mainly governed by weak intermolecular forces between adsorbate and adsorbent (e.g. Van der Waals or other electrostatic forces), and thus requires a lower binding energy than chemisorption [3]. This therefore means that physisorption requires a low temperature and high pressure. It is worth noting that more corrugated surfaces may result in more adsorption, as the corrugations are closer to the atoms on top of adding more effective area. Since the attraction increase by the inverse of the distance, this results in stronger binding sites compared to a smoother surface [4, 5]. Another difference with chemisorption is that it can accommodate multiple layers of adsorbate.

A fundamental way to characterize the adsorption behaviour of an adsorbate in an adsorbent is through an adsorption isotherm, i.e. a function of the loading as a function of pressure under a constant temperature. The most basic formulation of the isotherm is the one given by Langmuir [6]. For a fixed maximum loading q_{max}, a fitted pairwise constant b, the loading $q(P)$ as a function of pressure becomes

$$q(P) = q_{\text{max}} \frac{bP}{1+bP} \tag{1.1}$$

At low pressure (also referred at dilute region), the loading becomes linearly dependent on the pressure through Henry's law [7]

$$q(P) = K_H P \tag{1.2}$$

By combining Equation 1.1 and 1.2, one can obtain an expression for the loading solely dependent on the on the limits of the isotherm

$$q(P) = \frac{\frac{K_H}{q_{\text{max}}} P}{1 + \frac{K_H}{q_{\text{max}}} P} \tag{1.3}$$

If isotherms for pure compounds can be obtained, ideal adsorbed solution theory [8, 9] can be used to predict mixture-based properties, such as their loading or competitiveness of adsorption between different species.

In the context of maximum loading, physisorption dominates as this relates to the physical limit of adsorbent phase that can be accommodated. This means that critical factors limiting the former are mostly related to geometric constraint and surface areas available in the adsorbent.

1.1.2. Applications

The phenomenon of adsorption has many practical uses, but there are three applications that stand out: separation processes, storage and catalysis.
The first type is the one related to separation or purification. Since adsorption can be applied selectively on species, it is used to removes contaminants and impurities from gas and liquid streams [10, 11]. Examples of this application include carbon capture and storage [12] and waste water purification [13]. A common process using this is pressure-swing adsorption [14], where the pressure can be used to selectively adsorb species in gas mixes. This is commonly used in the purification of hydrogen, where it is used to separate the hydrogen gas during processes such as steam methane reforming [15].
Another application of adsorption processes is in storage. Because of the large internal areas of porous materials, molecules may be stored inside at lower pressures and temperatures than required. Two examples of compounds that can be stored in such manners are methane [16] and hydrogen, the latter being applicable for both combustible and fuel cell hydrogen. This same mechanism can also be used to store heat, by the adsorption and desorption of molecules in an adsorbate and using the heat of adsorption [17].
Depending on the adsorbent, adsorption can also help in catalysis. This can arise as they may contain active sites for catalysis (e.g. acid sites). A prominent example of this are zeolites, which can contain acid sites used for petrochemical reactions such as hydroisomerization or hydrocarbon cracking [18] whilst still offering species selectivity through their pore structure. Overall, adsorption can be used in a wide array of situations. Moreover, the tunability of adsorbent materials allows it to be specifically tailors for intended uses.

1.2. Zeolites

Zeolites are defined as microporous crystalline aluminosilicate materials, made up of a regular arrangement of tetrahedral TO_4 structures, composed of a tetrahedrally coordinated atom T (usually silicone or aluminium, denoted Si and Al respectively) bounded by four oxygen atoms. To differentiate them from other similar materials, the amount of T-atoms per 1000 [Å] is set at between 12 and 21 [19]. These collections of tetrahedral units form networks of pores and channels throughout the material, allowing them to offer a large surface area and volume all whilst being thermally stable.
So far, there are more than 250 individually topologies recognized by the International Zeolite Association, each being attributed a three-letter code [20]. If one counts the amount of zeolites that are thermodynamically stable enough to be synthesized, the number goes to potential millions [21].

Pure-silica zeolites have no net electrical charge. However, the inclusion of aluminium atoms (i.e. substitution of Si^{4+} atoms with Al^{3+} in the framework leads to a net negative charge [22]. In this case, extra-framework cations need to be present to balance the charges. This leads to zeolites being represented by the formula

$$\mathrm{M}_{\mathrm{y}}^{\mathrm{z+}}[\mathrm{Si}_{1\text{-}\mathrm{x}}\mathrm{Al}_{\mathrm{x}}\mathrm{O}_2]^{\mathrm{x-}} \qquad \mathrm{x} = \mathrm{y} \cdot \mathrm{z} \qquad 0 \leq \mathrm{x} \leq 0.5 \tag{1.4}$$

for a given number y of extra-framework cations M with valence z. It is important to note that the amount of Al-based tetrahedrons (and thus the amount and distribution of extra-framework cations) is controlled by Löwenstein's rule [23], which states that two Al atoms cannot share a common oxygen.
With the presence of extra-framework cations, zeolites can be used as catalysts [24]. This is because these create Brønsted and Lewis acid sites. Brønsted sites are able give H^+ ions to diffusing molecules whilst Lewis acid sites can accept electron pairs.

The main property of zeolites is that they are porous materials, meaning that they present voids in their structure not occupied by framework atoms [22]. The latter are referred as pores. This also implies the formation of channels, through which molecules may use to diffuse through the zeolite. Their size, connectivity, topology and geometry may therefore be used to describe the behaviour of molecules diffusing through the crystal. One such dimension is the minimum window size, being determined by the number of T-coordinated atoms forming the opening. This helps classing zeolites depending on their minimum pore diameter into small (3 to 5 [Å]), medium(3 to 6 [Å]), large (3 to 7.5 [Å]) and extra-large (above 7.5 [Å]) pore zeolites.

Another essential property of zeolites is their shape-based selectivity [25, 18]. Since molecules diffuse through the channels, it is evident that the channel system can influence what molecules can diffuse in them.
In the context of zeolites used as catalysts, shape-based selectivity is applicable to reactants, transition states and products alike [26]. Reactants that are too large to enter the channel network will not undergo the chemical process, thus favouring smaller molecules. The transition-states (or intermediate molecules) formed during the reaction, or can even be suppressed if the needed molecule, thus favouring alternative pathways. The product shape selectivity also disfavours large molecules, as they remain trapped inside the framework or may undergo other reactions (e.g. cracking in alkanes). This makes zeolites effective as molecular sieves, as well as controlling the distribution of products post-catalysis.
In the case of alkanes, these are called entropy effects [27], and come under three form: size, configurational and length entropy. The size entropy dictates that alkanes made of less carbons are preferred as they pack more efficiently. Configurational entropy states that linear alkanes are preferred over branched ones as they pack more efficiently. Lastly, length entropy states that when faced with cylindrical channel, branched alkanes with more compact shapes are preferred over linear alkanes with the same amount of carbons.

Combining this information with what was said in section 1.1, it is therefore evident that the leading factors in determining the maximum loading in full-silica zeolites are the channel topology and geometry, as well as the available volume and surface area. Futhemore, the alkane properties such as size and configuration will play a role due to selectivity effects.

1.3. Computing the Maximum Loading

There are two types of molecular simulations: Molecular Dynamics and Monte-Carlo Simulation [28]. Focus will be on the latter (hereby referred as *MC*) as it is used in this work. The fundamental idea behind the *MC* algorithm in molecular simulations is first introduced. This is followed by an explanation of additional algorithms needed to compute the maximum loading for the work that is to follow.

1.3.1. Basic Molecular Simulation

Carrying out experiments that provide enough insight in the molecular behaviour is difficult. As such, molecular simulation is used since it is a cheaper alternative that can provide better insight [29].
Molecular simulation works by creating an atomistic model, meaning that the atoms for both molecules (hereby referred to as “guest”) and framework (hereby referred to as “host”) are characterized by a set of positions, masses and interaction parameters and interactions sites [30]. For the sake of reducing computational load, frameworks can be assumed as rigid, whereas molecules are treated as flexible to explore as many conformations as possible. A periodic boundary condition is employed in simulations as to mimic an infinitely big system.
Force fields are used to describe guest-guest and guest-host interaction. Force fields are specialized functions that describe the potential energy landscape as a function of the atomic positions [30, 31]. They do so by means of individual contributions encompassing bonded (intramolecular) interactions, such as bond stretching, angle bends and torsions, and non-bonded interactions, including van der Waals or electrostatic interactions (e.g. Coulombic attraction). For both guest-guest and guest-host interactions, the values of the main parameters are taken from experiments or quantum calculations. It is to be noted that only interactions below a certain distance are computed explicitly by the force field. Anything beyond this distance (referred as “cut-off distance”) is approximated by means of a tail correction [28].
Statistical mechanics is used connect the various interactions and actual observed properties [28]. It provides a base on which the macroscopic behaviour (property over the ensemble) can be determined based on the microscopic behaviour (individual behaviour of molecules or atoms). It does so making a probability distribution of microscopic states based on their energies and then makes an average based on what states are most likely to appear (i.e. those with lower energies).

1.3.2. The Monte-Carlo Algorithm

The core idea behind Monte Carlo simulation is the law of large numbers [32]. It states that if an arbitrary observable system property A can be expressed as an ensemble average, over a probability distribution $p(x)$, then the estimator

$$\langle A \rangle \approx \frac{1}{N} \sum_{i=1}^{N} A(x_i) \tag{1.5}$$

converges to the true expected value as the number or cycles N becomes infinitely large. In the case of molecular simulation [28], these system properties averages are of the type

$$\langle A \rangle = \frac{\int d\mathbf{r}^N \exp[-\beta \mathcal{U}(\mathbf{r}^N)] A(\mathbf{r}^N)}{\int d\mathbf{r}^N \exp[-\beta \mathcal{U}(\mathbf{r}^N)]} \tag{1.6}$$

with $\mathbf{r}^N$ the 3N-dimensional configuration vector expressing the position of all particles, $d\mathbf{r}^N$ an infinitesimal change imposed to the system to explore a new configuration, $\mathcal{U}(\mathbf{r}^N)$ the potential energy of a configuration dictated by $\mathbf{r}^N$, $A(\mathbf{r}^N)$ the value of the observable. To ensure that the most likely that the more likely energy states are emphasized, a Boltzmann parameter $\beta = \frac{1}{k_B T}$ is also used. It is to be noted that the denominator is also the partition function Z of the system, used to not only compute the thermodynamical properties of the system, but also obtain the probability distribution function $\mathcal{N}$ of configurations

$$\mathcal{N}(\mathbf{r}^N) = \frac{\exp[-\beta \mathcal{U}(\mathbf{r}^N)]}{Z} \tag{1.7}$$

Since all of these configurations cannot be all evaluated at once due to the amount of configurations of atoms growing exponentially with each additional atom, a sampling strategy needs to be set up.

To do so, the Metropolis method is used. This method makes use of Markov chains, in which the probability of visiting a new configuration can be computed from the current one. To make it easier to understand, the subscripts o and n are used to denote the old and new configurations of the system.

Given an old system configuration $\mathbf{r}_o^N$ with Boltzmann weight $\exp[-\beta\mathcal{U}(o)]$, the system can propose a new configuration $\mathbf{r}_n^N$ with weight $\exp[-\beta\mathcal{U}(n)]$ with a certain transition probability $\pi(o \to n)$. This includes moves such as translations and rotations. Since π must not destroy the distribution, the average number of accepted trial moves that leave state o should be balanced with the total amount of moves from all other states n to o. This means

$$\mathcal{N}(o) \sum_N \pi(o \to n) = \sum_N \mathcal{N}(n) \pi(n \to o) \tag{1.8}$$

π can therefore be interpreted as a transition matrix with the attached probabilities between states o and n describing the Markov process. To find the actual form of the transition matrix, it is decomposed into a matrix of trial moves $\alpha(o \to n)$ and a matrix of acceptances attached to these moves acc$(o \to n)$. Under the original assumption of acc$(o \to n)$ being symmetric, Equation 1.8 can be rewritten to

$$\mathcal{N}(o) \times \text{acc}(o \to n) = \mathcal{N}(n) \times \text{acc}(n \to o) \tag{1.9}$$

This means that

$$\frac{\text{acc}(o \to n)}{\text{acc}(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} \tag{1.10}$$

It is to be noted that acc$(o \to n)$ cannot be greater than 1. As such an upper bound is enforced on $\frac{\mathcal{N}(n)}{\mathcal{N}(o)}$.

The last step in the Markov chain is to see whether the moves is accepted or rejected. If the old and new configurations have potential energies $\mathcal{U}(o)$ and $\mathcal{U}(n)$ respectively ($\mathcal{U}(n) > \mathcal{U}(o)$), then the attached probability is determined by Equation 1.10. This probability is then compared to a random number $\mathcal{R}$, sampled from the uniform distribution (i.e. all outcomes have equal probabilities) in the $[0, 1]$ interval. The move is formally accepted if $\mathcal{R} <$ acc$(o \to n)$ and rejected otherwise. The move is always accepted in the inverse case ($\mathcal{U}(n) < \mathcal{U}(o)$).

1.3.3. Monte Carlo in Adsorption Computations

In the previous subsection, the general idea behind the Monte Carlo algorithm and its use of the Metropolis algorithm for sampling were explained with the assumption of three quantities being fixed: the number of particles, volume and temperature inside the system (also called constant-NVT ensemble). However, adsorption also calls for moves of particles insertion and deletions. As such, some changes need to be implemented in how moves are carried out. To solve this problem, adsorption simulations are carried out in the grand-canonical ensemble, with fixed chemical potential, volume and temperature (constant-μVT). The idea makes use of two systems: the system in which the averaging is carried out (denoted as system 1), and a reservoir (denoted as system 2). Based on the thermodynamics of the combined system, in which the condition for which the grand potential Ω, defined as

$$\Omega = U - TS - N\mu \tag{1.11}$$

to be minimized is if the chemical potential of both the reservoir and the considered system are equal. Probabilities for particle insertions $(N \rightarrow N+1)$ and deletions $(N-1 \rightarrow N)$ need to be defined for the Metropolis sampling method explained earlier. These take the form of

$$\text{acc}(N \rightarrow N+1) = \min\left[1, \frac{fV}{(N+1)}\exp[\mathcal{U}(N+1) - \mathcal{U}(N)]\right] \tag{1.12}$$

$$\text{acc}(N-1 \rightarrow N) = \min\left[1, \frac{V}{fV}\exp[\mathcal{U}(N-1) - \mathcal{U}(N)]\right] \tag{1.13}$$

with f the fugacity of the system

A second important improvement is the way the particles are inserted in the system. So far, the particles is inserted in one piece. This can be done for small particles, but becomes inconvenient when dealing with long-chain alkanes as they become very flexible. On top of this, the probability of successful insertion becomes very low at high pressures as can be seen in Equation 1.12. A further justification is that if molecules overlap on each other or the zeolite structure, their energy becomes ∞, and thus, the attached Boltzmann weight becomes close to zero.

A solution that can be deployed is the Configurational-Bias Monte Carlo algorithm [27, 28]. This algorithm builds the final particles one monomer at a time, biasing its generation to accommodate the present environment. This bias is then removed when computing the acceptance of the move.

The general procedure goes as following. First, the center of mass is slightly shifted, for which its orientation-dependent u_{or} energy is computed. Next, k trial orientations (individually denoted $b_j, j = \{1, 2, ..., k\}$) are generated, for which a position-dependent energy u_{pos} is associated. With those, the Rosenbluth factor

$$W(n) = \sum_{j=1}^{k} \exp[-\beta u_{or}(b_j)] \tag{1.14}$$

From there, a specific orientation n is selected, with probability

$$p(b_n) = \frac{\exp[-\beta u_{or}(b_n)]}{\sum_{j=1}^{k}\exp[-\beta u_{or}(b_j)]} \tag{1.15}$$

The Rosenbluth factor of the original configuration $W(o)$ is also computed, alongside $k-1$ trial orientations. The move (here the complete regrowth of the entire particle) has an attached acceptance of

$$\text{acc}(o \rightarrow n) = \min\left(1, \frac{W(n)}{W(o)}\exp\{-\beta[u_{pos}(n) - u_{pos}(o)]\}\right) \tag{1.16}$$

This allows to grow the molecule in the system whilst still obtaining the most energetically favourable molecule, whilst the Rosenbluth factor makes sure that the introduced bias is corrected for as not to affect the final sampling.

1.4. Machine Learning Algorithms

Machine Learning (abbreviated ML) is defined as a subfield of mathematics and computer science that uses algorithms to allow computers to learn from data without being explicitly programmed. These methods are best suited for situations where relationships between descriptors (inputs) and targets (outputs) are either too complex (e.g. highly non-linear, multi-dimensional, or analytically hard to derive) or too costly to obtain (e.g. lengthy simulations needed) [33].

The goal behind those algorithms is to create an approximation ($f : \mathbf{x} \to y$) that maps the underlying relationships between a set of values $\mathbf{x}$ and a set of outputs y. This approximation is parametrized by a set of parameters θ, that are optimized to minimize a loss function

$$\mathcal{L}(y, \hat{y}) \tag{1.17}$$

which quantifies the discrepancy between the real values of the target y and model-predicted values $\hat{y}$ (e.g. mean squared error or mean absolute error). Depending on the specific algorithm, optimization of the parameters is carried out using gradient-based methods, convex optimization, sequential fitting, or ensemble averaging. This procedure is referred to as the model ”learning” from the data. Alongside those learnable parameters, most models rely on a set of hyperparameters controlling the model structure and learning process. Examples include the tree depth in decision trees or the kernel width in support vector machines. Contrary to θ, model hyperparameters remain fixed during the learning process. As such, hyperparameter tuning is crucial to ensure a good generalization performance and low overfitting.

Since the inputs and output are both known, this work focuses on algorithms for supervised learning [33, 34]. Three types of commonly-used ML algorithms are considered in this work: tree-based algorithms, support vector machines, and deep learning algorithms. Tree-based methods construct hierarchical partition models by splitting through features and values. Since single trees are not reliable enough, methods use ensembles of trees such as random forests [35] or gradient-boosting [36] to increase the quality and robustness of prediction by either comparing predictions of individual trees or compensating prediction errors. Support Vector Machines (SVM) transform the input data into higher-dimensional feature spaces using kernel functions to learn non-linear boundaries using linear functions [33, 37]. Deep learning approaches use layers of neurons and leverage multiple non-linear transformations to learn highly complex functional mappings [38]. Practical examples include feedforward neural networks and transformer-inspired models such as the TABPFN model [39]. The strengths and weaknesses of each of the mentioned models are also elaborated upon.

1.4.1. Tree-based algorithms

Fundamental concept

Tree-based models work by repeatedly splitting the data into regions of homogeneous target values based on the values of features. To get the most heterogeneous split possible, a criterion is used depending on the task at hand. The variance is used for regression, and information gain or GINI impurity is used for classification [33].

The reduction in criterion ΔI between a parent (upstream) node containing N_{parent} samples and j child (downstream) nodes with N_j samples at a candidate split is defined as with variance in the parent set $I(\text{parent})$, and the individual variance of the child nodes $I(\text{child}_j)$ is defined as

$$\Delta I = I(\text{parent}) - \sum_j \frac{N_j}{N_{\text{parent}}} I(\text{child}_j), \tag{1.18}$$

Some hyperparameters need to be specified before the tree is trained in order to specify its structure, the most important of which are the number of leaves and the depth of the tree. In this context, a leaf is a node at the end of the tree, after which no split occurs. These allow the model to capture the finer details in the data, but may risk in overfitting if there are not checked. The depth of a tree is defined as the longest path between the root node and any leaf node. This helps the model capture complex behaviours. This hyperparameter is also vital as shallow trees might underfit by not capturing complex

relationships, and deep trees might start modelling noise.
Both hyperparameters, as well as imposing a minimum amount of datapoints for which a split can occur, are therefore used to control the complexity of the final model [33]. A visual depiction can be seen in Figure 1.1

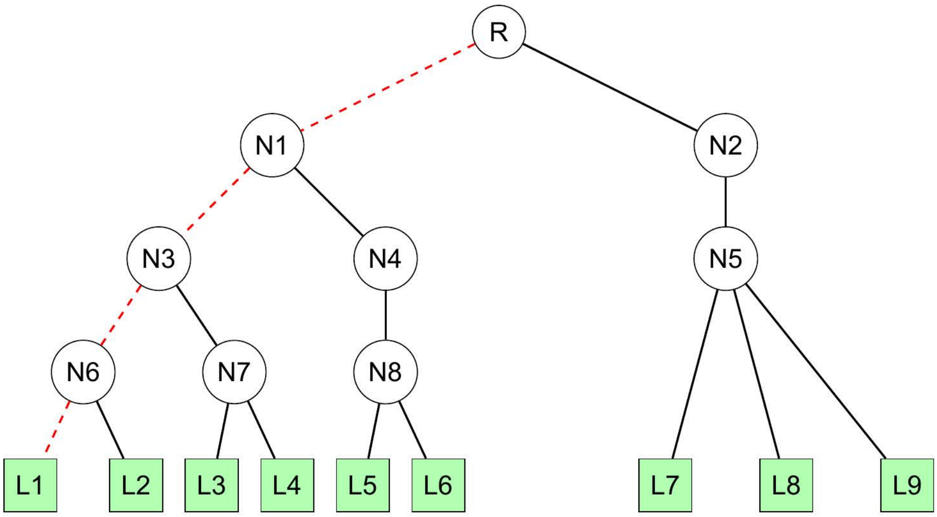


Figure 1.1: Arbitrary decision tree of 9 leaves (represented by the green squares) and depth 4 (represented by the red dashed edges).

One advantage of algorithms is that it can naturally handle nonlinear and inter-descriptor relationships without needing to scale features. Moreover, decision trees are more interpretable, thus promoting more transparency in the prediction process. However, single trees can be prone to overfitting. That can be the case if the variance in the data is too great or if some hyperparameters (such as the ones previously mentioned) are left unregulated. On top of this, trees are neither continuous nor smooth since they predict piecewise approximations.

Random forests

Since a single tree is prone to overfitting, a possibility is to combine multiple independently-trained trees, and combine their individual outputs by means of averaging all individual predictions. This leads to the random forest model [33, 35]. The central concept behind it is bootstrap aggregation, a method in which each individual tree is trained on a random subset of the data. Additionally, at each split in a tree, only a random subset of features is considered. This feature randomness reduces correlation among trees, thus increasing ensemble diversity and reducing overfitting.
For regression, the resulting prediction $\hat{y}$ of a random forest composed of T individual trees $f_t(\mathbf{x})$ in parallel is given by

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x}) \tag{1.19}$$

Each tree independently maximizes variance reduction at each node. The combination of averaging (also called ”bagging”) and feature subsampling helps making sure the model does not become too complex (also referred as ”regularization”), making random forests robust to noise in the training data.

This means that the number of trees is an important hyperparameter in random forests. The final performance can be increased by adding more trees, hence reducing the variance of the final averaging, albeit this improvement decreases beyond a certain point. As such, the number of tress should be balanced to offer optimal performance at low computational cost.
This means that an advantage of random forests is that they can still output good predictions even if a single tree overfits [40]. A drawback of random forests is that they have limited interpretability compared to single trees. Feature importance can still be estimated over the forest, but interpretability of individual trees is hard top piece due to the bagging step.

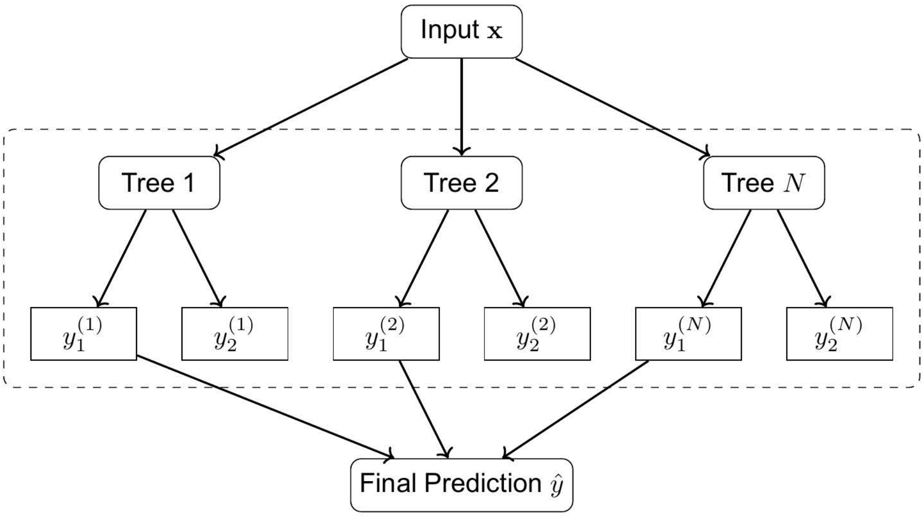


Figure 1.2: Schematic of the working principle behind random forests. An ensemble of decision trees are trained independently and their predictions are averaged to produce the final output.

Gradient boosting

Another way to improve on the original decision tree using ensembles is by using gradient boosting [36]. Unlike the random forest, this technique builds trees sequentially, with each new tree trained to correct the errors of the previous one.
To add a new error-correcting tree (m) to the ensemble, the pseudo-residuals r_m are computed as the negative gradient of the loss function $\mathcal{L}$ with respect to the current model $F_{m-1}(\mathbf{x_i})$

$$r_m = -\left[\frac{\partial \mathcal{L}(y_i, F_{m-1}(\mathbf{x}_i))}{\partial F_{m-1}(\mathbf{x}_i)}\right] \tag{1.20}$$

Using those, a new regressor tree model $h_m(\mathbf{x})$ is trained on r_m. The ensemble prediction $F_m(\mathbf{x})$ is updated by adding the tree, scaled by a learning rate λ

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \lambda h_m(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{k=1}^{m} \lambda h_k(\mathbf{x}) \tag{1.21}$$

Shallow trees are initially used as weak learners, which allows incremental improvements without overfitting.

Using this concept, Extreme Gradient Boosting enhances traditional gradient boosting by including second-order Taylor approximations of the loss function, as well as explicit regularization on the tree complexity [41]. It tries to find a pseudo-residuals tree that minimizes the second-order approximated objective

$$\sum_{i}^{N} \left[g_{im} * h_m(\mathbf{x}_i) + \frac{1}{2} * h_{im} * h_m(\mathbf{x}_i)^2 \right] + \Omega(h_m) \tag{1.22}$$

with

$$g_{im} = \frac{\partial \mathcal{L}(y_i, F_{m-1}(\mathbf{x}_i))}{\partial F_{m-1}(\mathbf{x}_i)} \qquad h_{im} = \frac{\partial^2 \mathcal{L}(y_i, F_{m-1}(\mathbf{x}_i))}{\partial (F_{m-1}(\mathbf{x}_i))^2} \qquad \Omega(h_m) = \gamma T + \frac{1}{2}\alpha \sum_{j=1}^{T} w_j^2, \tag{1.23}$$

where the $\Omega(h_m)$ term being dependent on the leaf penalty constant γ, the total number of leaves T, the leaf weight w_j and the leaf shrinkage α. This term in particular serves to regularize the tree and prevent overfitting, as γT penalizes the tree for excessive depth and leaves and $\frac{1}{2}\alpha \sum_{j=1}^{T} w_j^2$ makes sure that leaves do not become too dominant.

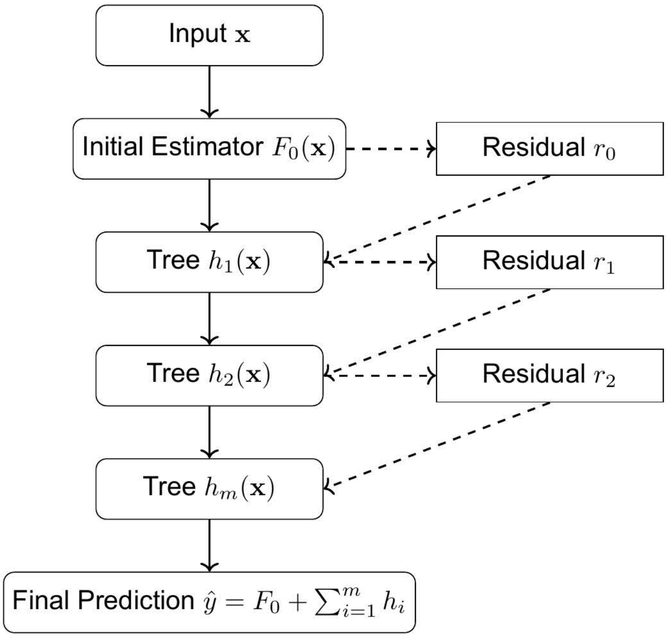


Figure 1.3: Schematic of the working principle behind gradient boosted trees. The initial estimator $F_0(\mathbf{x})$ provides a baseline prediction, and subsequent trees $h_i(\mathbf{x})$ are trained sequentially to model the residuals, gradually improving the prediction.

1.4.2. Support Vector Machines

Support Vector Machines (SVM) are algorithms that aim to find the best decision boundary in a high-dimensional space, occasionally with the help of kernel functions [33, 37]. If given a descriptor set $\mathbf{x}$ and a target set $\mathbf{y}$ with values bounded in the $[-1, 1]$ interval, a decision boundary $f(\mathbf{x} \to \mathbf{y})$ can be defined by

$$f(\mathbf{x}) = \beta_0 + \beta\phi(\mathbf{x}) \tag{1.24}$$

with a scalar intercept β_0, a weight vector in the feature space β and a mapping $\phi(\mathbf{x})$. $\phi(\mathbf{x})$ has the goal of producing the closest fit to the target within a tube of ε sensitivity.

If SVMs are used for classification, the decision boundary is obtained by means of an optimization problem formulated as [37]

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N} \xi_i, \tag{1.25}$$

with C a regularization parameter and a variable ξ to allow some minor violation of constraints as to accommodate non-separable data.
The optimization is also subjected to constraints

$$y_i(\beta^\top \phi(\mathbf{x_i}) + \beta_0) \geq 1 - \xi_i \qquad \xi_i \geq 0, \tag{1.26}$$

the latter being derived from the hinge loss function, helping to penalize points outside the given margin.
A drawback stemming from the formulation of Equation 1.25 is that it is very sensible to the values of β. As such, the data needs to be scaled as to not cause variables with large numeric ranges to dominate. A simple scaling may include standardizing the predictors to a distribution with zero mean and unit variance [34].

In the case of regression, the SVM (now referred as SVR) will seek a regression function that can remain within a window ε, and only starts penalizing when the absolute error gets above the latter. In this scenario, two slack variables ξ_i^+ and ξ_i^- are employed to quantify the over- and under-shoot from the ε tolerance respectively [37].

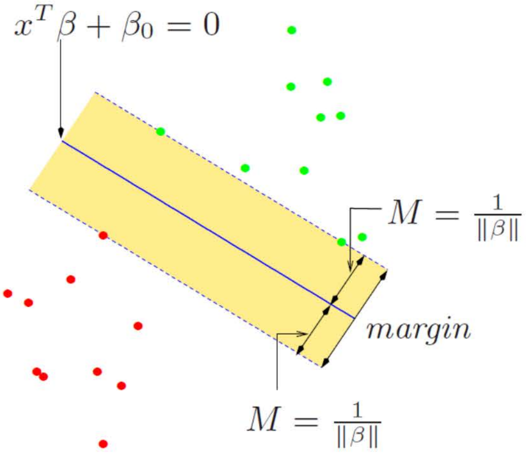


Figure 1.4: Visual depiction of the support vector classifier between the red and green classes [37]. The solid line represents the decision boundary, and the dashed lines show the maximal margin allowed for deviation.

As such, the optimization problem formulated in Equation 1.25 is transformed into

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}(\xi_i^- - \xi_i^+), \tag{1.27}$$

subjected to

$$y_i - f(\mathbf{x}_i) \leq \varepsilon - \xi_i^+ \qquad f(\mathbf{x}_i) - y_i \leq \varepsilon - \xi_i^+ \qquad \xi_i^-, \xi_i^+ \geq 0 \tag{1.28}$$

When dealign with regression, kernels can be used to determine how flexible the decision boundary can be. It can also be used to approximate $\phi(\mathbf{x})$ without needing to explicitly compute it, as the kernel K evaluates similarities between the points in space. This is done by the dot-product between two mappings $\phi(\mathbf{x_i})$ and $\phi(\mathbf{x_j})$ from two input vectors x_i and x_j

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \tag{1.29}$$

This is known as the kernel trick in SVMs.
When changed to accommodate the kernel trick, Equation 1.24 for the hyperplane becomes

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\alpha_i^- - \alpha_i^+)K(\mathbf{x}_i, \mathbf{x}_j) + \beta_0 \tag{1.30}$$

with α_i^- and α_i^+ being non-zero Lagrange multipliers for the lower and upper boundaries of an optimization.
Standard kernels for the regressor include linear, polynomial, Gaussian radial-basis function and sigmoid. Each have their own formula, tunable hyperparameters and effects on how they shape the decision boundary. All these properties are tabulated in Table 1.1 [37].

SVMs are effective on moderately sized datasets and high-dimensional feature spaces, providing robust generalization. On top of this, the optimization for determining the hyperplane allows for a framework robust against overfitting. Their limitations include poor scalability to large datasets as the operation requirement scales with the square of the amount of datapoints [33]. They also require careful kernel selection and subsequent hyperparameter tuning hyperparameter selection and interpretability is also limited compared to tree-based models as they neither provide probabilistic explanations or transparent insights in the decision-making process [40].

Table 1.1: Classical kernel types commonly used in SVRs, definition, tunable parameters and important properties. The kernel controls the mapping of the (non-)linear relationships in the high-dimensional space, thus controlling the smoothness and flexibility of the boundary function.

Kernel	Formula for $K(\mathbf{x}_i, \mathbf{x}_j)$	Tunable parameters	Notes
Linear	$\mathbf{x}_i^\top \mathbf{x}_j$	None	Simplest kernel; suitable when the relationship is approximately linear.
Polynomial	$(\gamma\, \mathbf{x}_i^\top \mathbf{x}_j + r)^d$	Degree d Scaling γ Offset r	Captures polynomial interactions up to degree d. Can be high-variance for large d.
Gaussian RBF	$\exp\big(-\gamma \lVert \mathbf{x}_i - \mathbf{x}_j \rVert^2\big)$	Width (scaling) γ	Most common default for non-linear problems. Small γ yields smooth, global fits; large γ yields localized, flexible fits.
Sigmoid	$\tanh(\gamma\, \mathbf{x}_i^\top \mathbf{x}_j + r)$	Scaling γ Offset r	Resembles a two-layer neural network activation.

1.4.3. Deep learning algorithms

A drawback of machine learning is that data needs some degree of pre-processing before being fitted to a model (e.g. the transformation required for SVM usage mentioned in subsection 1.4.2). This means that processing natural data in their raw form (e.g. including noise or missing values) is difficult [42]. To mitigate this, deep learning methods use multiple levels of representation and abstraction of data. Their architectures stack layers of nonlinear transformations, allowing them to learn representations that could normalize, de-noise or restructure data as needed. This allows them to bypass steps that would be used in ML such as feature engineering or data transformation.

When speaking of deep learning, neural networks are generally the most popular model. These models share similarities with how the brain work, by connecting layers of neurons between each other. Other models do exist (such as Boltzmann machines or auto-encoders), but are out of scope in this work.

Neural Networks

As mentioned, the neural network takes inspiration on the human brain, by creating a model composed of multiple interconnected layers made out of neurons [38]. The most basic neural network that be created is the feed-forward artificial neural network. It is constituted of 3 types of layers: an input layer (taking in the inputs), the output layer (giving the actual prediction) and hidden layers (layers between input and output that carry out the operations via activation functions). For a network with L layers, with each layer having neurons with an activation function σ and trainable weights $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$, the mathematical representation of each layer $\mathbf{h}^{(l)}$ becomes

$$\mathbf{h}^{(l)} = \sigma\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) \qquad l = \{1, 2, ..., L-1\} \tag{1.31}$$

with the output layer giving the value of the prediction $\hat{y} = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}$.

An important part of the operations in layers is the activation function σ, governing how neurons process the values they are fed. They then pass the values to the neurons in the next layer, also known as forward propagation. Standard activation functions for neurons include the linear, sigmoid, hyperbolic tangent (tanh) and rectified linear unit functions (abbreviated ReLU) [43]. The latter, defined as

$$\sigma(x) = \max(0, x) \tag{1.32}$$

has become popular for being able to handle issues of vanishing gradients in neural networks.

Beyond the classical neuron layers presented above, neural networks may implement convolutional layers [38]. These are especially handy in processing spatial structures such as images or 3D data. The underlying operation employed in this setting is convolution. This is done by feeding a input feature map

$\mathbf{X} \in \mathbb{R}^{H \times W \times d}$ through filters $\mathbf{K}^{(m)} \in \mathbb{R}^{k \times k \times d}$ using a discrete convolution operation. This operation, producing an output feature map $\mathbf{Y}^{(m)}$ is defined as

$$\mathbf{Y}_{i,j}^{(m)} = \sum_{u=1}^{k} \sum_{v=1}^{k} \sum_{c=1}^{d} \mathbf{K}_{u,v,c}^{(m)} \mathbf{X}_{i+u-1,\, j+v-1,\, c} + b^{(m)} \tag{1.33}$$

for a specified kernel width k working on d input channels, with (i, j) spatial position and a learnable bias $+b^{(m)}$ for the output channel m. After the convolution, the resulting map is fed to the activation function through

$$\mathbf{H}^{(\mathbf{m})} = \sigma(\mathbf{Y}_{i,j}^{(m)}) \tag{1.34}$$

Between convolution layers, specialized pooling layers work by aggregating over set spatial neighbourhoods, thus reducing both dimensionality and memory requirement, as well as introducing variance to small translations and rotations.
Other types of neural networks exist, but these are out of scope for this work.

The training procedure involves optimizing the weights in Equation 1.31 to fit the training data with respect to a specified loss function (usually mean squared error in the case of regression). The most-used method is known as backpropagation, which updates the weights of each neuron by starting from the output layer and going back layer by layer for set number of iterations. During this process, the network adjusts the parameters (denoted as $\theta \in \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$) by using gradient of the loss function with respect to each parameter $\frac{\partial \mathcal{L}(y,\hat{y})}{\partial \theta}$. The typical update of a weight $\mathbf{W}_{(i,j)}$ in a neural network with a predefined learning rate λ is of the form

$$\theta \leftarrow \theta - \lambda \frac{\partial \mathcal{L}(y, \hat{y})}{\partial} \tag{1.35}$$

By repeatedly updating these weights through each iteration, the network gradually reduces the value of the loss function by improving its predictions. A pseudo-algorithm of the process is demonstrated in 1.

Algorithm 1 Backpropagation for a feedforward neural network

```
Require: Input $\mathbf{x}$, target $y$, learning rate $\lambda$, parameters $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$
Ensure: Updated parameters $\theta$
Forward pass:
for layer $l = 1$ to $L - 1$ do
    $\mathbf{h}^{(l)} \leftarrow \sigma\big(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\big)$
end for
Output: $\hat{y} \leftarrow \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}$
Compute loss: $\mathcal{L}(y, \hat{y})$
Backward pass:
Compute output layer error: $\delta^{(L)} \leftarrow \frac{\partial \mathcal{L}}{\partial \hat{y}}$
for layer $l = L - 1$ down to $1$ do
    $\delta^{(l)} \leftarrow (\mathbf{W}^{(l+1)})^\top \delta^{(l+1)} \odot \sigma'(\mathbf{h}^{(l)})$
end for
Parameter update:
for layer $l = 1$ to $L$ do
    $\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \lambda\, \delta^{(l)} (\mathbf{h}^{(l-1)})^\top$
    $\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \lambda\, \delta^{(l)}$
end for
```

Using neural networks presents both advantages and disadvantages [44, 45]. It is evident that the main strength behind neural networks is their ability to handle much more complex data than the methods presented so far. It also extends at feature extraction, allowing them to determine which values are most relevant.However, this is only possible if computational resources are available, as training may require parallel processing. Furthermore, the use of layers and neurons make the entire model less interpretable both for explanation and debugging.

TabPFN Model

The TabPFN model (acronym for Tabular Prior-data Fitted Network) is a transformer-based model that is pre-trained on millions of synthetic regression and classification tasks, specifically designed for tabular data [39]. Because of this, it requires no tuning of hyperparameters.

TabPFN works by making Bayesian-style predictions, meaning they try to predict the most likely outcome for new inputs and their uncertainty. This is done by approximating the posterior predictive distribution (PPD), expressing the probability of an output $\hat{y}_{\text{new}}$ for a new input x_{new} given a training dataset $D_{train} = \{(x_i, y_i)\}_{i=1}^{n}$ by means of the integral

$$p(\hat{y}_{\text{new}}|x_{\text{new}}, D_{\text{train}}) \propto \int_{\Phi} p(\hat{y}_{\text{new}}|x_{\text{new}}, \phi)\, p(D_{\text{train}}|\phi)\, p(\phi)\, d\phi \tag{1.36}$$

In this context, Φ is the space of hypothetical operations ϕ that link x_{new} and $\hat{y}_{\text{new}}$.

During training, the TabPFN does not observe real datasets. Instead, it is trained once on synthetic datasets generated from the prior over operations $\phi \sim p(\phi)$. Each synthetic dataset D_{train} consists of feature–target pairs (x_i, y_i) produced by a sampled mechanism ϕ. The network learns to predict held-out target from these datasets by minimizing the cross-entropy loss

$$L_{\text{PFN}} = \mathbb{E}_{D \sim p(D)}\big[- \log q_\theta(y_{\text{test}}|x_{\text{test}}, D_{\text{train}})\big], \tag{1.37}$$

where q_θ denotes the TabPFN's predicted probabilities. Through this process, the network effectively approximates the integral in the PPD, learning to produce Bayesian-style predictions for any dataset consistent with the prior.

When predicting test values, the trained TabPFN can take any real tabular dataset D_{train} and a set of test inputs x_{test} and produce posterior predictive probabilities $q_\theta(y|x_{\text{test}}, D_{\text{train}})$ in a single forward pass. This makes TabPFN extremely fast, as no gradient-based learning or hyperparameter tuning is needed during this pass to format to training data. Additionally, the Transformer architecture allows it to handle variable-sized training sets and queries, and zero-padding ensures compatibility with datasets of different feature dimensions.

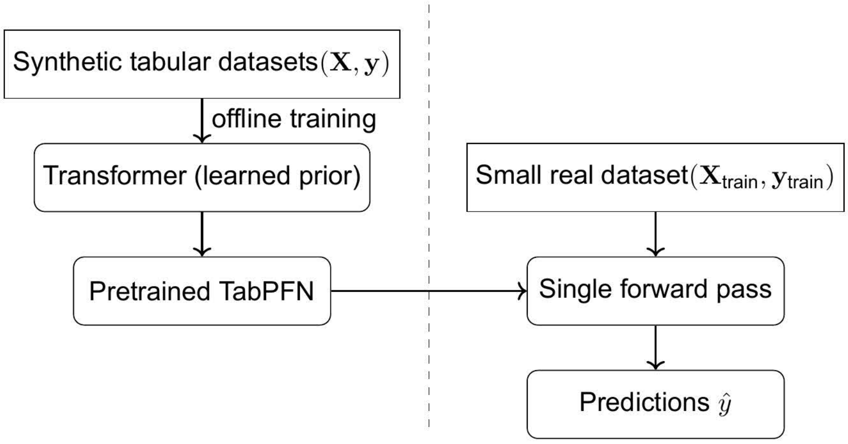

Synthetic tabular datasets$(\mathbf{X}, \mathbf{y})$
offline training
Transformer (learned prior)
Pretrained TabPFN
Small real dataset$(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$
Single forward pass
Predictions $\hat{y}$

Figure 1.5: Conceptual workflow of TabPFN. The model is pretrained offline on many synthetic tabular datasets to learn a prior over functions. At inference time, predictions are obtained via a single forward pass needing to tune hyperparameters or retrain the model.

Advantages of TabPFN include its ability to make the ability of making uncertainty-aware predictions with fast inference times. Its training method of using a large base of synthetic tasks makes it also robust to small or imbalanced datasets. And comparable to neural networks, it can also capture simple causal patterns in data without explicit feature engineering. Limitations of the model includes the network's performance depends on how well the synthetic prior reflects real-world tabular data, with extremely large or highly specialized datasets possibly falling outside the scope of patterns learned during the pre-training phase. Furthermore, as with all deep models, interpretability of predictions can be challenging, and the network may not perfectly capture complex dependencies not represented in the prior.

1.5. Current state of research and research gaps

ML algorithms have been used in porous media research for many purposes, but most of them are centred around one of two main uses: High-Throughput Computational Screening (HTCS) and Quantitive Structure-Property Relationships (QSPR).

HTCS applications revolve around using ML to find a candidate structure given a certain target quantity, albeit most of this type of work has been observed with Metallic-Organic Frameworks (MOFs). Xue et. al. [46] proposed using random forests to find the best MOF to separate propane and propylene. Zhang et al. [47] applied a similar method to screen the best-performing MOFs for methanol-ethanol separation. In the case of zeolites, Daou et. al. [48] used an algorithm to find the best candidates for adsorption of alkanes ranging from methane to hexane in silica zeolites, and achieved high performance with all molecules, albeit it degraded as molecules grew larger.

QSPR is used to try and predict a certain target quantity from a set of given variables. A few examples include Evans et. al. [49], who used gradient-boosted trees to find the bulk and shear moduli of zeolites using local geometry, structure and porosity information. Some approaches involving neural nets include ZeoNet3D [50], that uses convolutional neural networks for predicting Henry coefficients, or Petcovic et. al. [51], who used graph neural networks for not only predicting Henry coefficients and heats of adsorption, but also used the model for inverse design purposes (i.e. obtain geometry or topology through optimization if the output is known) to get the optimal zeolite topology for a given heat of adsorption. There are existing ML attempts at predicting the maximum loading of molecules in zeolites. An example is Wu et. al. [52] used looked at multiple ML algorithms to try and predict the maximum loading of CO_2 for wastewater treatment and found that gradient-boosted trees worked best, with a similar conclusion drawn by Li et. al. [53], who tried to predict the maximum adsorption capacity of methane in coal.

Most of the works mentioned above use models relying mostly on geometrical and topological descriptors to predict the selected target. Some works used the heat of adsorption to predict the maximum loading, such as Daou et. al. [48] that used the Henry coefficient and the heat of adsorption at zero loading.

To summarize, the available literature about using ML in porous media is present, even when looking for the maximum loading specifically. Most works tend to either use random forests or gradient-boosted trees since they are easier to implement and already show good performance. Some directly use deep-learning approaches, allowing them to extract more complex representations of features, but the relative performance improvement from ML is minimal [54]. Furthermore, the main types of descriptors used are of the geometrical and topological type. Some works do include chemical descriptors, although they are irrelevant to determining maximum loading. Energy descriptors have also been used (such as the heat of adsorption), but these require prior molecular simulations, thus undermining the goal of ML to be a possible replacement tool.

From the examples of literature above, it can be seen that there are a few research gaps when looking at the use of ML in porous media adsorption, namely:

1. **Lack of work involving long-chain and branched alkanes**
   So far, most of the research using ML in porous media is concerned with molecules such as methane, water or CO_2. These are much smaller than the alkanes generally found in petroleum fuel. A first reason for this is that the majority of research in adsorbates focuses on using them as sieves or scrubbers for industrial processes, in which most of these molecules are present. The second reason is that smaller molecules are computationally lighter to deal with in simulation. Moreover, only one or two adsorbates are considered at a time. And even so, they tend to have separate models depending on species. This results in some possible important adsorbent-adsorbate relationships possibly being missed. Lastly, most work involving alkanes does not include branched isomers because large molecules present the risk of activity cliffs [55, 56]. These activity cliffs represent a significant challenge for ML models because they assume a smooth landscape between input and output.
2. **Lack of work bridging the gap between predictions and physics**
   Some of the works mentioned above are carried out with the mind of reaching the highest prediction performance possible. However, they do not always attempt to explain whether the reasoning

behind the model actually aligns with the physics involved. This is because ML models are by themselves incapable of understanding the underlying physics [57]. This is further blurred if neural networks are involved [58]. Tools such as SHapely Additive exPlanations (SHAP) do allow to see what descriptors have the most weight in model predictions [59], but this is not standard procedure.

1.6. Research questions

The previous section outlined the current state of research in porous materials, with an emphasis on predicting the maximum loading using ML algorithms. It was concluded that despite work having been done, there were gaps. Specifically, that work so far did not consider large alkanes in zeolites, as well as focusing more on the prediction performance of the model rather than its physical correctness.
From there, an overarching research question may be formulated, being

How can the maximum loading of long-chain alkanes in zeolites be efficiently and accurately predicted using machine learning?

A first reason why this research question is relevant is because the alkane-zeolite combinatorial space is immense. There are roughly 261 catalogued zeolites with millions of potentially feasible structures [48, 60]. Moreover, the number of alkane isomers grows uncontrollably for every added carbon atom [61]. Tools like molecular simulation can only sample one combination at a time, thus creating the need for something faster and cheaper. Another reason is that this question can help pave the way for more complex calculations. So far, understating of adsorption is constrained to pure components. However, one can deal with the adsorption of mixtures via Ideal Adsorbed Solution Theory (IAST) if an expression for the adsorption isotherm can be obtained [9]. This makes it possible to study competitive adsorption between species, hence gaining insight on what species can and cannot be obtained or diffuse through the channels. Equation 1.3 shows that if one has the Henry coefficient and the maximum loading, an approximate for the isotherm can be obtained with the Langmuir model. Since a ML model already exists for the Henry coefficient [56], the maximum loading therefore becomes the last piece of puzzle.
A further set of secondary research questions is also formulated as to address certain key aspects of the main question. These include

What descriptors can be used to represent long-chain alkanes in machine learning-based adsorption predictions?

How can meaningful physical relationships between long-chain alkanes and zeolites be modelled and learned by machine learning algorithms?

Which machine learning algorithm offers the best balance between accuracy, interpretability and robustness for predicting adsorption properties in zeolites?

These questions are also important as adsorption is a complex phenomenon with many possible variables. And as stated in section 1.5, ML models tend not to capture the physical relationships in the data unless it is explicitly coded in the data. Considering that one of the research gaps in hydroisomerization is the lack of an optimal zeolite topology for a given process [29], finding what the most influential variables dictating the adsorption capacity is important. This is relevant since zeolites have have the property of shape-based selectivity. This therefore affects the products distribution post-hydroisomerization, which may lead to different properties than planned.
Together, these questions determine not only the scope of the study, but also the foundation on which the methods and following results are obtained.

References

[1] Matthias Thommes et al. “Physisorption of gases, with special reference to the evaluation of surface area and pore size distribution (IUPAC Technical Report)”. In: *Pure and Applied Chemistry* 87.9-10 (July 2015), pp. 1051–1069. DOI: 10.1515/pac-2014-1117. URL: https://doi.org/10.1515/pac-2014-1117.

[2] Tawfik Abdo Saleh. *Surface and morphological characterization of hybrid materials*. Jan. 2021, pp. 241–283. DOI: 10.1016/b978-0-12-813294-4.00003-0. URL: https://doi.org/10.1016/b978-0-12-813294-4.00003-0.

[3] Mohammad Ismail et al. *Novel materials and technologies for hydrogen storage*. Jan. 2020, pp. 337–365. DOI: 10.1016/b978-0-12-819553-6.00014-3. URL: https://doi.org/10.1016/b978-0-12-819553-6.00014-3.

[4] L.W. Bruch. “Theory of physisorption interactions”. In: *Surface Science* 125.1 (Feb. 1983), pp. 194–217. DOI: 10.1016/0039-6028(83)90453-3. URL: https://doi.org/10.1016/0039-6028(83)90453-3.

[5] Eda Gökırmak Söğüt and Mehmet Gülcan. *Adsorption: basics, properties, and classification*. Elsevier, Jan. 2023. ISBN: 9780443184567. DOI: 10.1016/B978-0-443-18456-7.00001-8.

[6] Irving Langmuir. “The adsorption of gases on plane surfaces of glass, mica and platinum”. In: *Journal of the American Chemical Society* 40 (9 Sept. 1918), pp. 1361–1403. ISSN: 15205126. DOI: 10.1021/ja02242a004.

[7] Jianlong Wang and Xuan Guo. “Adsorption isotherm models: Classification, physical meaning, application and solving method”. In: *Chemosphere* 258 (June 2020), p. 127279. DOI: 10.1016/j.chemosphere.2020.127279. URL: https://doi.org/10.1016/j.chemosphere.2020.127279.

[8] A. L. Myers and J. M. Prausnitz. “Thermodynamics of mixed□gas adsorption”. In: *AIChE Journal* 11 (1 1965), pp. 121–127. ISSN: 15475905. DOI: 10.1002/aic.690110125.

[9] C. J. Radke and J. M. Prausnitz. “Thermodynamics of multi□solute adsorption from dilute liquid solutions”. In: *AIChE Journal* 18.4 (July 1972), pp. 761–768. DOI: 10.1002/aic.690180417. URL: https://doi.org/10.1002/aic.690180417.

[10] Eduardo Antonio Pinto Dias et al. *CO2 adsorption by zeolites: State-of-Art, techniques and emerging trends*. Jan. 2026. DOI: 10.1016/j.micromeso.2025.113904.

[11] Haoxin Mai et al. *Machine Learning in the Development of Adsorbents for Clean Energy Application and Greenhouse Gas Capture*. Dec. 2022. DOI: 10.1002/advs.202203899.

[12] Federica Raganati, Francesco Miccio, and Paola Ammendola. “Adsorption of carbon dioxide for post-combustion capture: A review”. In: *Energy Fuels* 35.16 (Aug. 2021), pp. 12845–12868. DOI: 10.1021/acs.energyfuels.1c01618. URL: https://doi.org/10.1021/acs.energyfuels.1c01618.

[13] Mohammad Hadi Dehghani et al. “Recent advances on sustainable adsorbents for the remediation of noxious pollutants from water and wastewater: A critical review”. In: *Arabian Journal of Chemistry* 16.12 (Sept. 2023), p. 105303. DOI: 10.1016/j.arabjc.2023.105303. URL: https://doi.org/10.1016/j.arabjc.2023.105303.

[14] Elshaday Mulu, Milton M. M’Arimi, and Rose C. Ramkat. “A review of recent developments in application of low cost natural materials in purification and upgrade of biogas”. In: *Renewable and Sustainable Energy Reviews* 145 (Apr. 2021), p. 111081. DOI: 10.1016/j.rser.2021.111081. URL: https://doi.org/10.1016/j.rser.2021.111081.

[15] Mauro Luberti and Hyungwoong Ahn. “Review of Polybed pressure swing adsorption for hydrogen purification”. In: *International Journal of Hydrogen Energy* 47.20 (Feb. 2022), pp. 10911–10933. DOI: 10.1016/j.ijhydene.2022.01.147. URL: https://doi.org/10.1016/j.ijhydene.2022.01.147.

[16] A. Celzard and V. Fierro. “Preparing a suitable material designed for methane storage: A comprehensive report”. In: *Energy Fuels* 19.2 (Feb. 2005), pp. 573–583. DOI: 10.1021/ef040045b. URL: https://doi.org/10.1021/ef040045b.

[17] Salvatore Vasta et al. “Adsorption Heat Storage: State-of-the-Art and Future Perspectives”. In: *Nanomaterials* 8.7 (July 2018), p. 522. DOI: 10.3390/nano8070522. URL: https://doi.org/10.3390/nano8070522.

[18] Shrinjay Sharma et al. “Prediction of Thermochemical Properties of Long-Chain Alkanes Using Linear Regression: Application to Hydroisomerization”. In: *Journal of Physical Chemistry B* (2024). ISSN: 15205207. DOI: 10.1021/acs.jpcb.4c05355.

[19] Christian Baerlocher, Lynne B. McCusker, and David H. Olson. *Atlas of Zeolite Framework Types*. 6th ed. Elsevier, 2007, pp. 3–11. ISBN: 9780444530646. DOI: 10.1016/B978-0-444-53064-6.X5186-X.

[20] Ch. Baerlocher et al. *Database of Zeolite Structures*. https://www.iza-structure.org/databases/. maintained by the Structure Commission of the International Zeolite Association (IZA□SC). 2025.

[21] R. Pophale, P. A. Cheeseman, and M. W. Deem. “A Database of New Zeolite□Like Materials”. In: *Phys. Chem. Chem. Phys.* 13 (2011), pp. 12407–12412. DOI: 10.1039/C0CP02255A.

[22] Eduardo Pérez-Botella, Susana Valencia, and Fernando Rey. *Zeolites in Adsorption Processes: State of the Art and Future Prospects*. Dec. 2022. DOI: 10.1021/acs.chemrev.2c00140.

[23] W. Löwenstein. “The Distribution of Aluminum in the Tetrahedral Framework of Silicates and Zeolites”. In: *American Mineralogist* 39 (1954), pp. 92–96. DOI: 10.2138/am-1954-39-92.

[24] Smt Sami Almutairi. “The role of Lewis and Brønsted acidity for alkane activation over zeolites”. In: *Data Archiving and Networked Services (DANS)* (Jan. 2013). DOI: 10.6100/ir755379. URL: https://research.tue.nl/nl/publications/9454e4a4-c06a-420f-bd1f-89cce2e901e6.

[25] Yuanlong Han et al. “Shape selectivity of zeolite for hydroisomerization of long-chain alkanes”. In: *New Journal of Chemistry* 47 (3 Dec. 2022), pp. 1401–1412. ISSN: 13699261. DOI: 10.1039/d2nj04976g.

[26] Jens Weitkamp. *Catalysis and Zeolites*. Ed. by Jens Weitkamp and Lothar Puppe. Vol. 131. Springer Berlin Heidelberg, 1999, pp. 175–188. ISBN: 978-3-642-08347-1. DOI: 10.1007/978-3-662-03764-5. URL: http://link.springer.com/10.1007/978-3-662-03764-5.

[27] Rajamani Krishna, Berend Smit, and Sofia Calero. “Entropy effects during sorption of alkanes in zeolites”. In: *Chemical Society Reviews* 31 (3 2002), pp. 185–194. ISSN: 03060012. DOI: 10.1039/b101267n.

[28] Berend Smit and Daan Frenkel. *Understanding Molecular Simulation*. 3rd ed. Elsevier, 2002. ISBN: 9780122673511. DOI: 10.1016/B978-0-12-267351-1.X5000-7. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780122673511X50007.

[29] Berend Smit and Theo L.M. Maesen. “Molecular simulations of zeolites: Adsorption, diffusion, and shape selectivity”. In: *Chemical Reviews* 108 (10 Oct. 2008), pp. 4125–4184. ISSN: 00092665. DOI: 10.1021/cr8002642.

[30] David Dubbeldam, Ariana Torres-Knoop, and Krista S. Walton. “On the inner workings of Monte Carlo codes”. In: *Molecular Simulation* 39.14-15 (Oct. 2013), pp. 1253–1292. DOI: 10.1080/08927022.2013.819102. URL: https://doi.org/10.1080/08927022.2013.819102.

[31] Peng Bai, Michael Tsapatsis, and J. Ilja Siepmann. “TrAPPE-ZEO: Transferable potentials for phase equilibria force field for All-Silica Zeolites”. In: *Journal of Physical Chemistry C* 117.46 (Oct. 2013), pp. 24375–24387. DOI: 10.1021/jp4074224. URL: https://doi.org/10.1021/jp4074224.

[32] John. Dagpunar. *Simulation and Monte Carlo : with applications in finance and MCMC*. John Wiley, 2007, p. 333. ISBN: 9780470854945.

[33] Sarah Guido and Andreas C Mueller. *Introduction to machine learning with python*. O’Reilly Media, 2016, p. 392. ISBN: 9781449369415.

[34] Aurélien. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2019, p. 819. ISBN: 9781492032649.

[35] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[36] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29 (5 Oct. 2001). ISSN: 0090-5364. DOI: 10.1214/aos/1013203451. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full.

[37] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7. URL: http://link.springer.com/10.1007/978-0-387-84858-7.

[38] Josh Patterson and Adam Gibson. *Deep Learning A Practitioner's Approach*. O'Reilly, 2017. ISBN: 978□1491914250. URL: http://oreilly.com/safari.

[39] Noah Hollmann et al. "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second". No journal attached. Sept. 2023. arXiv: 2207.01848 [cs.LG]. URL: http://arxiv.org/abs/2207.01848.

[40] Ozan Kocadagli et al., eds. *Proceeding Book of the y-BIS Conference 2019: Recent Advances in Data Science and Business Analytics*. Istanbul, Turkey: y-BIS Conference, Sept. 2019. ISBN: 978-605-5005-95-5.

[41] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". No jounral attached. June 2016. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[42] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. *Deep learning*. May 2015. DOI: 10.1038/nature14539.

[43] Abien Fred Agarap. "Deep Learning using Rectified Linear Units (ReLU)". In: *arXiv (Cornell University)* (Mar. 2018). DOI: 10.48550/arxiv.1803.08375. URL: http://arxiv.org/abs/1803.08375.

[44] Elham Kariri et al. "Exploring the Advancements and Future Research Directions of Artificial Neural Networks: A Text Mining Approach". In: *Applied Sciences (Switzerland)* 13 (5 Mar. 2023). ISSN: 20763417. DOI: 10.3390/app13053186.

[45] Maad M. Mijwel. "Artificial Neural Networks: Advantages and Disadvantages". In: *Mesopotamian Journal of Big Data* 2021 (2021), pp. 29–31. DOI: 10.58496/MJBD/2021/006. URL: https://mesopotamian.press/journals/index.php/bigdata/article/view/225.

[46] Xiaoyu Xue et al. "High-Throughput Screening of Metal-Organic Frameworks Assisted by Machine Learning: Propane/Propylene Separation". In: *Industrial and Engineering Chemistry Research* 62 (2 Jan. 2023), pp. 1073–1084. ISSN: 15205045. DOI: 10.1021/acs.iecr.2c02374.

[47] Lulu Zhang et al. "Automatic Machine Learning Combined with High-Throughput Computational Screening of Hydrophobic Metal-Organic Frameworks for Capture of Methanol and Ethanol from the Air". In: *ACS ES and T Engineering* 4 (1 Jan. 2024), pp. 115–127. ISSN: 26900645. DOI: 10.1021/acsestengg.2c00424.

[48] Alan S.S. Daou et al. "Machine Learning and IAST-Aided High-Throughput Screening of Cationic and Silica Zeolites for Alkane Capture, Storage, and Separations". In: *Journal of Physical Chemistry C* 128 (14 Apr. 2024), pp. 6089–6105. ISSN: 19327455. DOI: 10.1021/acs.jpcc.4c00066.

[49] Jack D. Evans and François Xavier Coudert. "Predicting the Mechanical Properties of Zeolite Frameworks by Machine Learning". In: *Chemistry of Materials* 29 (18 Sept. 2017), pp. 7833–7839. ISSN: 15205002. DOI: 10.1021/acs.chemmater.7b02532.

[50] Peng Liu. *Bayesian Optimization: Theory and Practice Using Python*. Apress, 2023. ISBN: 978-1-4842-9062-0. DOI: 10.1007/978-1-4842-9063-7.

[51] Marko Petković et al. “Graph Neural networks for carbon dioxide adsorption prediction in Aluminium-Exchanged zeolites”. In: *arXiv (Cornell University)* (Mar. 2024). DOI: 10.48550/arxiv.2403.12659. URL: http://arxiv.org/abs/2403.12659.

[52] Haibin Wu et al. “Based on machine learning model for prediction of CO2 adsorption of synthetic zeolite in two-step solid waste treatment”. In: *Arabian Journal of Chemistry* 17 (2 Feb. 2024). ISSN: 18785352. DOI: 10.1016/j.arabjc.2023.105507.

[53] Wenshuo Li et al. “Machine Learning Algorithm to Predict Methane Adsorption Capacity of Coal”. In: *Energy and Fuels* 38 (24 Dec. 2024), pp. 23422–23432. ISSN: 15205029. DOI: 10.1021/acs.energyfuels.4c04906.

[54] Saerom Park et al. “Machine learning-based prediction of adsorption capacity of metal-doped and undoped activated carbon: Assessing the role of metal doping”. In: *Chemosphere* 366 (Oct. 2024). ISSN: 18791298. DOI: 10.1016/j.chemosphere.2024.143495.

[55] Derek Van Tilborg, Alisa Alenicheva, and Francesca Grisoni. “Exposing the Limitations of Molecular Machine Learning with Activity Cliffs”. In: *Journal of Chemical Information and Modeling* 62 (23 Dec. 2022), pp. 5938–5951. ISSN: 1549960X. DOI: 10.1021/acs.jcim.2c01073.

[56] Shrinjay Sharma et al. “Machine Learning-Based Predictions of Henry Coefficients for Long-Chain Alkanes in One-Dimensional Zeolites: Application to Hydroisomerization”. In: *The Journal of Physical Chemistry C* 129 (40 Oct. 2025), pp. 18234–18249. ISSN: 1932-7447. DOI: 10.1021/acs.jpcc.5c03868.

[57] Arijit Chakraborty et al. “Explainable AI modeling of zeolite adsorption isotherms”. In: *Chemical Engineering Science* 320 (Jan. 2026). ISSN: 00092509. DOI: 10.1016/j.ces.2025.122361.

[58] Rishi Gurnani et al. “Interpretable Machine Learning-Based Predictions of Methane Uptake Isotherms in Metal-Organic Frameworks”. In: *Chemistry of Materials* 33 (10 May 2021), pp. 3543–3552. ISSN: 15205002. DOI: 10.1021/acs.chemmater.0c04729.

[59] Christoph. Molnar. *Interpretable machine learning : a guide for making black box models explainable*. 2nd ed. Christoph Molnar, 2022, p. 317. ISBN: 9798411463330.

[60] *Hypothetical Zeolites Database*. Accessed: 2025-12-11. URL: http://www.hypotheticalzeolites.net/.

[61] S. R. Rieder et al. “Development of an open-source software for isomer enumeration”. In: *Journal of Cheminformatics* 15 (2023), p. 10. DOI: 10.1186/s13321-023-00683-2.

Predicting the Maximum Loading in Zeolites for Hydroisomerization Applications: A Machine Learning Approach

Eric Johnsson,[†] Shrinjay Sharma,[‡] Arvind Gangoli Rao,[†] David Dubbeldam,[¶] Sofia Calero,[§] and Thijs J. H. Vlugt[*,‡]

[†]Flight Performance & Propulsion Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

[‡]Process & Energy Department, Faculty of Mechanical Engineering, Delft University of Technology, Leeghwaterstraat 39, 2628CB Delft, The Netherlands

[¶]Van 't Hoff Institute of Molecular Sciences, University of Amsterdam, Science Park 904, 1098XH, Amsterdam, The Netherlands

[§]Department of Applied Physics, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands

E-mail: t.j.h.vlugt@tudelft.nl

Abstract

Hydroisomerization of alkane isomers is an important step in the manufacture of current kerosene and sustainable aviation fuels. Zeolites are used as acid catalysts in the process. It is therefore important to have predictions of the maximum loading of hydrocarbons in zeolites. Here, a cascade model using machine learning models is used to predict the maximum loading of alkane isomers in zeolites. The cascade is composed of a gradient-boosted tree classifier stage that predicts whether adsorption occurs or not, and a regressor predicting the value of the maximum loading. The final dataset consists of 45 different molecules (both linear and branched alkanes up to C_{16}) and 97 different zeolites structures, resulting in 4365 datapoints. Descriptors include information on the geometry and topology of zeolite channels, as well as shape and size of molecules. Extra composite descriptors are also present to provide the models a physical basis for predictions. Multiple regressors of different nature are considered: Support Vector Regressors, Gradient-Boosted Trees, extreme Gradient-Boosted Trees, and the TabPFN pretrained model. Out of all the models, TabPFN yields the highest generalization performance and lowest error. An interpretability analysis is conducted to assess whether the decisions abide by the governing physics of adposition. It is confirmed that the top descriptor choices abided by the necessary physical constraints, but also that secondary properties such as shape-based selectivity are also accounted for. It is shown that despite both classifier and regressor being insensitive to random splits in data, the regressor is prone to overfitting at low fractions of data withheld for testing. The cascade model is compared with an Artificial Neural Network for training and deployability. Despite training taking more resources for the neural network, the latter is lighter both in memory and storage when compared to the cascade. This work builds on previous research in predicting the Henry coefficient at zero loading. Using this previous model and the findings of this work, one can draw the full adsorption isotherm for any alkane, thus enabling the analysis of adsorption behaviour of alkane mixtures using IAST.

1. Introduction

Climate change has been one of the most pressing issues in recent years, being the subject of international agreements.[1] This is pushing fossil-fuel dependent industries to research more sustainable practices. For the aviation industry (which accounts for about 2.5% of global carbon emissions[2]), solutions to reduce the industry footprint without sacrificing passenger demand are to either re-design components, or to try and switch to a more environmentally-friendly fuel.[3] One such fuel is Sustainable Aviation Fuel (SAF),[3,4] made from non-fossil sources such as fats[5] (e.g. used cooking oil or animal fats), biomass fuels[3] or syngas from fossil sources (e.g. coal or natural gas). SAF promises to significantly reduce carbon emissions while still being compatible with current engines and fuel infrastructure (both for processing and distribution).[3]

The most commercially viable way to make SAF is through the HEFA pathway (Hydroprocessed Esters and Fatty Acids),[3] which transforms natural oils and fats into usable jet fuel. This pathway uses a two-fold process of deoxygenation and hydroisomerization.[3] The latter, also know as catalytic de-waxing,[6] transforms linear alkane chains into branched isomers, by means of a bi-functional catalyst composed of a metal (generally platinum) and a porous crystalline material called a zeolite. Linear alkanes are first delivered to the metal site, where transformation into olefins occurs. These olefins then diffuse to the zeolite, where Brønsted acid sites are encountered, changing these olefins into carbenium ions. Subsequent deprotonation is carried out, resulting in branched alkenes. By a final step of hydrogenation, the alkenes are transformed into branched alkanes.[7] Making alkanes undergo hydroisomerization offers multiple benefits when looking at fuel performance. First, it improves cold-flow properties (e.g. viscosity and freezing point).[8] Second, it increases the octane rating of the fuel, making it less prone to ignite under compression.[8]

Despite the current understanding on hydroisomerization, important research gaps such as competitive adsorption or optimal zeolite topology for a given process still need to be addressed.[9] Carrying out experiments that can provide sufficient information at the molecular level are difficult to set up. Experimental hydroisomerization studies also tend to focus on single components only, meaning that effects of competitive adsorption and desorption are largely overlooked. As a cheaper alternative to adsorption experiments, molecular simulations using Monte Carlo simulations in the grand-canonical ensemble can be performed.[10] Despite these advancements, a larger problem has yet to be addressed: the large combinatorial space between alkanes and zeolites. The International Zeolite Association (IZA) recognizes 261 synthesizable zeolite structures to date, and the number goes to millions when considering hypothetical structures.[11,12] The number of possible alkanes also explodes for large numbers of carbon atoms.[13] This renders current methods such as molecular simulation very inefficient for screening purposes due to the attached time and computational cost. There is thus a need of a fast, cheap, and reliable method to predict adsorption properties of alkanes isomers in zeolites. An effort has already been made by predicting the Henry coefficient K_H that describes the adsorption at infinite dilution.[14] If one assumes a single-site Langmuir-type behaviour, an expression for the full isotherm can be made with both the Henry coefficient and the maximum loading.[15] These two quantities also serve as inputs for Ideal Adsorbed Solution Theory (IAST)[16–18] for studying adsorption properties of complex alkane mixtures.[19] Therefore, there is a need for a way to compute the maximum uptake of long-chain alkane in zeolites.

To address the problem of the large combinatorial space, Machine Learning (ML) algorithms have become an increasingly popular way, thanks to the data-driven approach. This allows to capture complex relations between a set of inputs (called descriptors) and outputs (called targets), all whilst being less expensive than classical molecular simulations or experiments. ML algorithms have been extensively used in research relating to porous materials in recent years.[20–22] Some of their most prevalent uses are as tools for High-Throughput Computational Screening (HTCS) or for Quantitive Structure-Property Relationships (QSPR). As highlighted in the reviews by Alitintas *et al.*[23] and Yang *et al.*[22] on the use of ML with Metal-Organic Frameworks (MOFs), ML is capable of using data pertaining to pore size and geometry to screen structures for the best candidate if given a specific property. Xue *et al.*[24] used a random forest model to find the most viable candidates for the separation of propane and propylene amongst pure-ilica zeolites. An example of ML used for QSPR include the analysis by Xiuying *et al.*[25] to find the governing structural parameters behind the adsorption selectivity of CO_2 and N_2 in all-silica zeolites. Tatlier *et al.*[26] reported on the relation between the water uptake in zeolites and the structural and chemical properties of the framework. ML models can also be used to directly predict a large set of diverse properties. Evans *et al.*[27] used extremely gradient-boosted trees to determine mechanical properties such as the shear and bulk moduli of zeolites, providing insight into what quantities impact these moduli the most. Another work is the one of Yu *et al.*[28] to predict the henry coefficient of diverse molecules inluding hydrogen and CO_2 in MOFs. Attempts to predict maximum loadings in porous materials have been carried out in the past, albeit only for small molecules like methane or water or carbon dioxide. Some notable works include the work of Li et al.[29] on predicting the maximum loading of methane in coal, and Zhao *et al.*,[30] who used an Artificial Neural Network using structural and energy descriptors to predict maximum loadings of propylene in zeolites. Some works also address the maximum loading by predicting the entire adsorption isotherm, such as that of Chakraborty[31,32] on the full isotherms of methane, CO_2 and N_2 in zeolites. Despite the substantial amount of literature available on the topic of adsorption in porous materials, little research so far has been performed on using such ML methods for large alkanes, both linear and branched, in zeolites. One reason for that is the focus so far being gas separation and filtering applications involving smaller adsorbates (such methane, water, and CO_2).

In this work, a modelling framework using ML models to predict the maximum loading of both linear and branched alkanes in zeolites is presented to address this gap. The proposed approach uses two models in a series arrangement (otherwise known as a cascading model). This model splits the task of computing the into two smaller tasks. First, a classification predicting whether adsorption for a given alkane can fit in the zeolite structure as a binary outcome. If adsorption can occur, a regression is made to estimate the value of the maximum loading. The classification aspect is handled by a gradient-boosted tree classifier (GBC).[33,34] This model is based on a collection of decision trees in series where the next tree learns to correct the errors of the previous one.[33] Then, a separate stage composed of a regressor assesses the values of the maximum loading if it indeed occurs. Multiple models are considered, including support vector regressors (SVR),[33,35] gradient-boosted tree regressors (GBR),[33,34] extreme gradient-boosted regressor (XGB)[36] and the TABPFN model,[37] which is a pre-fitted transformer model. Except for TABPFN, all models are also sensible to hyperparameters, such as the learning rate (strength of error correction) for the GBC or the ε parameter (prediction tolerance) in the SVR. To ensure the highest performance under any dataset, hyperparameters are determined through Bayesian optimisation.

To generate training and testing data for the maximum loadings, the molecular simulation software RASPA2[38,39] is used. RASPA2 is an open-source software to compute adsorption and diffusion in porous materials such as zeolites using force field-based molecular simulation. To compute adsorption isotherms, it uses Monte Carlo simulations in the grand-canonical (μVT) ensemble using molecular interaction from the TrAPPE and TrAPPE-Zeo[40] force fields. With these and the available Configurational-Bias Monte-Carlo (CBMC) algorithm to generate conformations of large alkanes, RASPA2 provides accurate predictions for adsorption isotherm data.[41,42] In this work, only alkanes up to 16 carbon atoms (both linear and branched) are considered since these are the largest alkanes encountered in kerosene.[43] Furthermore, only side groups going up to 3 carbon atoms (i.e. methyl, ethyl, propyl and isopropyl groups) are considered. An important limitation on the analyzed zeolites is that only full-silica frameworks are considered. This leads to a final population of 45 alkanes and 97 zeolites respectively, resulting in total 4365 datapoints.

The paper is structured as follows. In Section 2, the methods used to gather the necessary data (both for the individual components and for the maximum loading), to create the model, and to evaluate its performance are explained. In Section 3, the model performance, interpretability, and robustness are discussed, and a comparison with a deep learning approach is presented. In Section 4, concluding remarks are provided together with a description of possible improvements and future work.

This article also contains a Supporting Information. SI1 is a pdf file containing the list of all zeolites and alkanes considered in this work, as well as the optimal model hyperparameter values and raw results of the robustness study. SI2 is an excel sheet acting as a log of all RASPA2 simulations (results and parameters). SI3 is a folder containing all the Python codes used for this work.

2. Methodology

Zeolite and Alkane Selection

The computational space for possible simulations is too large for all possible alkane-zeolite combinations. As such, some degree of filtering is needed to limit the number of computations and to make the simulations more manageable. Alkanes for this work are selected from a database from previous work.[14] This includes a complete list of all isomers from $n - C_1$ to $n - C_{20}$. Since the primary focus of this paper is SAF, the scope is restricted to alkanes containing up to 16 carbon atoms, corresponding to the largest molecules typically present in kerosene. Furthermore, work is limited to alkanes with methyl, ethyl, propyl and isopropyl side groups as alkanes with larger groups would not be able to diffuse in the zeolite channels.[14] A base set of alkanes comprising linear alkanes from methane to nonane, as well as some small isomers such as isobutane are selected to form a base set of selected molecules. Larger alkanes and isomers are selected using a random sampling method from a list of all hydrocarbons with a given carbon number, with a bias favouring molecules that are more different than the currently selected ones. Each molecule was represented as a feature vector $\mathbf{x}$ comprising of the main-chain carbon count, the total carbon number, the amount of side-groups as well as the share of carbons not included in the main chain. For a candidate molecule A (represented by vector $\mathbf{A}$), its structural similarity to each alkane B (vector $\mathbf{B}$) in the molecular set was quantified using the Tanimoto distance[44] defined as

$$D_T(A,B) = 1 - \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|^2 + \|\mathbf{B}\|^2 - \mathbf{A}\cdot\mathbf{B}}. \tag{1}$$

This is used to quantify how close the molecule is compared to another. Only when the average distance $\bar{D_T}$ is above 0.3 for all already selected alkanes, the molecule is added to the set. Otherwise, a new candidate is sampled from the database of all isomers. This technique is used to bring the total number of selected molecules to 45. The full list of alkanes can be found in the Supporting Information.

The zeolite data is obtained by combining the IZA online database,[45] the iRASPA visualization software,[46] and the structures found in the Atlas of Zeolite Structures.[47] From these three databases, zeolite structures are selected on the basis of four criteria:

1. **Orthorhombic unit cells**: To limit the computational load and improve simulation speed, only orthorhombic unit cells are considered. This is because using non-orthorhombic cells require intermediate steps (such as matrix inversions for computing distances[38,39]), and are thus more computationally expensive.

2. **Low unit-cell density within a supercell**: A rule of thumb in RASPA2 is that the computational time increases by the square of the amount of unit cells. Hence, larger units cells are preferred. The threshold for this work is defined at 20 unit cells in a domain that is at least twice the default cut-off radius of 12.8 [Å] in each direction of the used force field.

3. **Limited or no enclosed voids**: In GCMC simulations, a molecule is inserted where the volume allows it. This means that a molecule can be inserted in a void that is disconnected from the main channel system, which is not physically possible in reality. Methods to block these pockets do exist, but it has been chosen to exclude the most extreme cases such as CFI-type or IRN-type zeolites.

4. **Non-zero accessible volume ($\boldsymbol{AV}$)**: A subset of zeolite structures lacks continuous channels or interconnected pore networks, resulting in negligible accessible volume. Since such frameworks do not support meaningful transport or adsorption behaviour, these are excluded from further consideration.

Out of the 232 structures common to all considered databases, 97 satisfy the above requirements. The complete list of used alkanes and zeolites can be found in the Supporting Information.

Computing the maximum loading

The maximum loading of each alkane-zeolite combination is determined by a two-step process. The first step addresses whether adsorption of the alkane is possible in a given framework. For this, the ZEO++ tool is used.[48] Given a certain probe radius for a given molecule, ZEO++ can determine whether channels in the framework are accessible or not. If no channels are accessible, it is because the minimum cross-sectional size of the molecule (determined from the minimum ellipsoid diameter obtained in RDKit[49]) cannot be accommodated by the restricting pore diameter (RPD) of the zeolite channel. As such, the maximum loading $q_{\max}$ is set to zero for this particular pair. If channels are accessible, ZEO++ generates a block-pockets file to wall off possible inaccessible pockets in the structure. The maximum loading is then computed in RASPA2 using these files to prevent molecules being generated in inaccessible pockets. All simulations are performed using Monte Carlo simulations and are carried out in the grand-canonical (μVT) ensemble at a fixed temperature of 298 [K]. To ensure that maximum loadings are encountered, all simulations are carried out at an external pressure of 10^{10} [Pa]. An exception is made for simulations of methane (10^{14} [Pa]) due to its smaller size. The fugacity coefficient of the simulations is set to one to approximate ideal-gas conditions. The TrAPPE and TrAPPE-Zeo force fields are used to describe the intramolecular bonded (bond, bending and torsion) and non-bonded (Lennard-Jones) interactions, as well as guest-guest and guest-host interactions (Lennard-Jones). The zeolite structures are treated as rigid since structural changes during adsorption processes have a minor effect on adsorption.[50] A cut-off radius of 12.8 [Å] is used for describing Lennard-Jones potentials with tail corrections applied.[10] The alkanes are represented using the united-atom approach (i.e. carbon and hydrogens are represented as a single interaction site), and are inserted in the system using the Configurational-Bias Monte Carlo algorithm.[10,51] The ideal gas Rosenbluth weights of all considered molecules are computed in separate NVT simulations At high loadings, the acceptance probability of insertion and deletion moves becomes low. In such cases, ensemble averages are no longer representative of the maximum loading. Therefore, the mode of the maximum loadings is used to have a representative result whilst still retaining statistical robustness. To this end, simulations run for 2×10^6 Monte Carlo cycles to obtain sufficient statistics for a reliable number.

A possible source of errors is the size of the simulation box. Molecular simulations can suffer from finite-size effects, in which too small systems do not accurately depict the thermodynamic limit. This is best explained by Fig. 1. In a simulation box with a fixed amount of unit cells through which channels pass through (depicted in black), the maximum loading corresponds to the largest integer number of molecules (approximated as red rectangles) that can physically fit in the channel. Since the domain is finite, there is a discrete amount of possible "slots" the molecule can be placed in. This has the consequence of artificially-induced stepwise behaviour in the ensemble average, which can be mitigated by adding more unit cells in the relevant direction. For this reason, multiple box sizes are used, starting from the smallest possible cell until either a cap on the amount of unit cells or a maximum number of unit-cell expansions is reached. This implies that the size of each simulation box is proper to each alkane-zeolite combination. As such, a complete simulation log dictating simulation parameters (e.g. size of the simulation box, simulation time, convergence of finite-size effects, ...) and values of the resulting maximum loading is available in the Supporting Information.

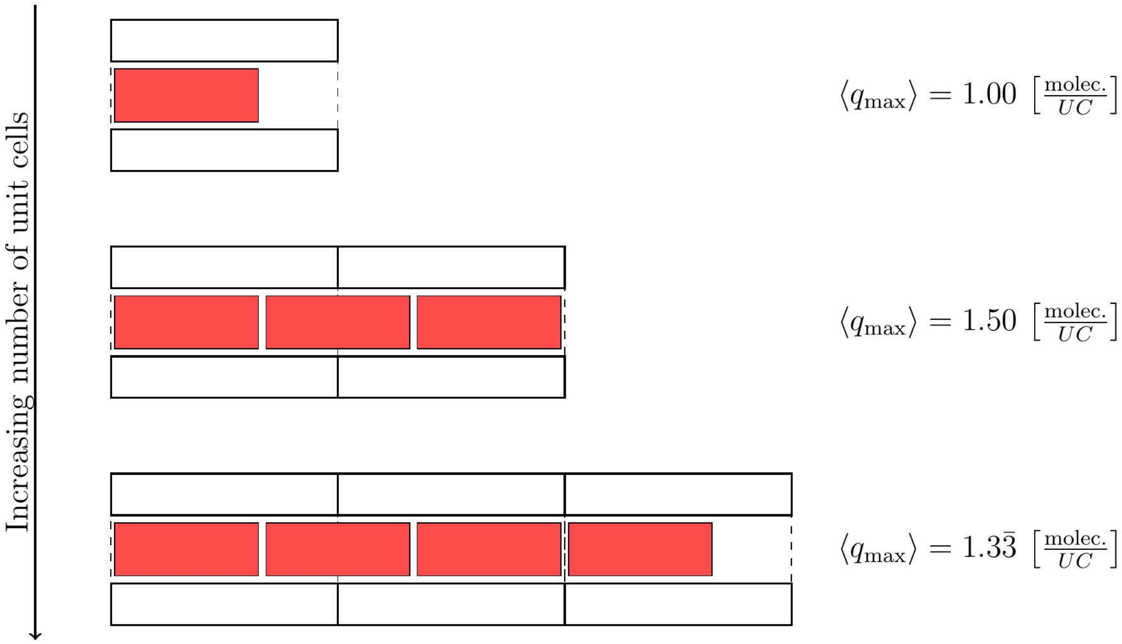

Increasing number of unit cells
$\langle q_{\max} \rangle = 1.00 \left[\frac{\text{molec.}}{UC}\right]$
$\langle q_{\max} \rangle = 1.50 \left[\frac{\text{molec.}}{UC}\right]$
$\langle q_{\max} \rangle = 1.3\bar{3} \left[\frac{\text{molec.}}{UC}\right]$

Figure 1: Visual depiction of the finite-size effects for adsorption of chain molecules. As both available space in the zeolite and the size of the molecule are fixed, the maximum loading of possible molecules physically fitting in the channel is capped to an integer amount. This can result in situations where leftover volume may be present, and thus influencing the final ensemble average of the maximum loading.

Descriptor Engineering and Preprocessing

To characterize different alkanes and zeolites, a set of quantities is chosen to describe individual properties as well as interactions with each other. Since the maximum loading is governed by physisorption, most descriptors used are of a geometrical or topological in nature. In this study, zeolites and alkanes are both characterized independently using 13 descriptors each, with an additional 3 descriptors to introduce physics in the dataset. It is intuitive that the maximum loading is dependent on the volume, surface area, and topological properties of a given framework. Topological descriptors include framework density (FD) and topological density on 10-membered rings (TD_{10}). To compute the volume and surface properties of the zeolites, the framework is imported in ZEO++.[48] Methane was selected as the probe instead of helium as to provide a more realistic estimate of the volumes and surfaces available to the molecules, since the smaller diameter of helium will incorrectly count hard-to-reach or inaccessible pores. The void fraction was computed using RAPSA2, where methane molecules are inserted according to the Widom particle insertion method.[10] Basic alkane descriptors are computed using the Python RDKit toolbox.[49] This includes molecular descriptors such as molar mass, amount of side groups of each type, total and main-chain carbons counts and molecular volume. As it is known that zeolites exhibit shape-based selectivity,[7,52,53] an emphasis on shape-based descriptors is made. This includes the acentric factor ω, which is computed by the method developed in the Supporting Information of Ref.[14] Another essential descriptor codifying the shape are the principal mass moments of inertia. Since RASPA2 generates a Boltzmann distribution of conformers of a molecule, the inertia tensor $\mathbf{I}$ can be built. From these, the values of the principal mass moments of inertia can be obtained by computing the eigenvalues of the tensor.

As mentioned earlier, the main reason molecules are not adsorbed in the zeolite is because of steric hindrance. As such, an essential dimension is an effective diameter for the molecule. Consequently, a proxy is derived by encapsulating the molecule within an ellipsoid to obtain an approximate of the size. Since larger molecules tend to be more flexible, this process is repeated for a large number of conformers created in the idea gas phase with RASPA2. The energies of each of these conformers is computed from the UFF[54] energy calculator in RDKit so that results can be Boltzmann-weighted. The UFF calculator computes the internal potential of an isolated molecule. This method is used to obtain the minimum and maximum diameters of the resulting ellipsoid, for which the mol-ellipsoid package[55] is used.

To enhance the information given by the descriptors, meaningful ratios can be formulated to emphasize interactions between the two. One of these values is roughness factor RI of zeolite channels, defined as

$$RI = \frac{AS}{AV^{\frac{2}{3}}} \tag{2}$$

in which ASA an AV are the methane-probed available surface and available volume inside the zeolite. The motivation behind it is that dimples in the channels can result in preferential adsorption sites for some molecules. Composite descriptors are also introduced to capture physical and mutual interaction relationships, thus creating a hybrid machine learning model that is physics-informed on top of being data-driven.[32] An example of this is the ratio χ between the minimum molecular diameter and the restricting pore diameter

$$\chi = \frac{D_{\text{Min}}^{\text{Mol}}}{RPD} \tag{3}$$

This helps encode the main steric relationship directly in the data, which would help the model understand it directly. It is using the same reasoning that a packing factor between zeolite and molecular volumes

$$\Psi = \frac{AV}{V_{\text{Mol}}} \tag{4}$$

is introduced. As maximum loading is physisorption dependent, multiple layers of adsorbate can form, which means that volume relationship between adsorbate and adsorbent becomes important since it constitutes a physical constraint. A summary table of all descriptors used for the models can be seen in table 1.

Both the maximum loading and the descriptors naturally span multiple orders of magnitude. This can result in skewed distributions which may result weak performance from the models due to the high variance.[33,56] This step is especially necessary for the Support Vector Machine due to its reliance on distances to target data points.[35] As such, a check of the data spread needs to be carried out. This can be addressed by making a probability distribution of the normalized values of the maximum loading, as shown in Fig. 2. It is clearly observed that in non-transformed conditions, the data presents a right-tail skew, i.e. most values cluster on the lower end with only a few reaching the high end. Since the unusually large maximum loadings can be observed for special cases of methane in large-pore zeolites, and larger molecules generally show lower loadings, a logarithmic transformation is applied to compress the dynamic range of the data. This operation reduces the relative influence of high-capacity outliers while expanding the spread to smaller values, thereby improving

Table 1: Table containing all zeolites, molecules and composite descriptors used in the model. Values for the zeolites were obtained from Refs.,[47][45] and.[46] Values for the molecules were obtained with RDkit.[49] The definitions of RI, χ and Ψ can be found in Eqs. 2 3 and 4

Descriptor	Category	Symbol	Units
Void fraction	Zeolite	ϕ	$[-]$
Methane-Accessible Area	Zeolite	ASA	m^2/g
Methane-Accessible Volume	Zeolite	AV	cm^3/g
Overall Accessible Volume	Zeolite	$AccV$	%
Methane-Non-accessible Area	Zeolite	$NASA$	m^2/g
Methane-Non-accessible Volume	Zeolite	NAV	cm^3/g
Framework Density	Zeolite	FD	g/cm^3
Topological Density (10-membered rings)	Zeolite	$TD10$	–
Channel dimension	Zeolite	d	Å
Restricting Pore Diameter	Zeolite	RPD	Å
Largest Cavity Diameter	Zeolite	LCD	Å
Molar Mass	Zeolite	MM	g/mol
Gravimetric Density	Zeolite	ρ	g/cm^3
Molecular Weight	Alkane	MW	g/mol
Main Chain Length	Alkane	MCL	# of carbons
Total Carbon Count	Alkane	TCC	# of carbons
Side Group Composition	Alkane	$[n_{\mathrm{C}}, n_{\mathrm{C2}}, n_{\mathrm{C3}}, n_{\mathrm{iC3}}]$	counts
Acentric Factor	Alkane	ω	–
Molecular Volume	Alkane	V_M	Å^3
Minimum Ellipsoid Diameter	Alkane	$D_{\mathrm{Min}}^{\mathrm{Mol}}$	Å
Maximum Ellipsoid Diameter	Alkane	$D_{\mathrm{Max}}^{\mathrm{Mol}}$	Å
Principal Mass Moments of Inertia	Alkane	PMI_i, $i = \{1, 2, 3\}$	$\mathrm{amu}\cdot\text{Å}^2$
Ratios of first and second Principal Mass Moments of Inertia	Alkane	$\frac{PMI1}{PMI2}$	$[-]$
Ratios of second and third Principal Mass Moments of Inertia	Alkane	$\frac{PMI2}{PMI3}$	$[-]$
Roughness Index	Composite	RI (see Eq. 2)	$[-]$
$\frac{D_{\mathrm{Min}}^{\mathrm{Mol}}}{RPD}$	Composite	χ (see Eq. 3)	$[-]$
$\frac{AV}{V_{\mathrm{Mol}}}$	Composite	Ψ (see Eq. 4)	$[-]$

the sensitivity of the model for the full range of maximum loadings.[57] As such, the final transformation carried out on the maximum loading is given by

$$q_{\max}^{*} = \log_{10}\left(1 + \frac{q_{\max}}{q_0}\right) \tag{5}$$

with q_0 being a unit maximum loading at 1 $\left[\frac{\mathrm{molec.}}{\mathrm{m}^3}\right]$ to make the argument inside the logarithm dimensionless. This results in a distribution that resembles more closely a bimodal distribution with the possibility of each peak being addressed separately, thus making it less prone to biases. This helps preserving the physical meaningfulness of the zero-values of adsorption in the final dataset. The same transformation is carried out for highly skewed

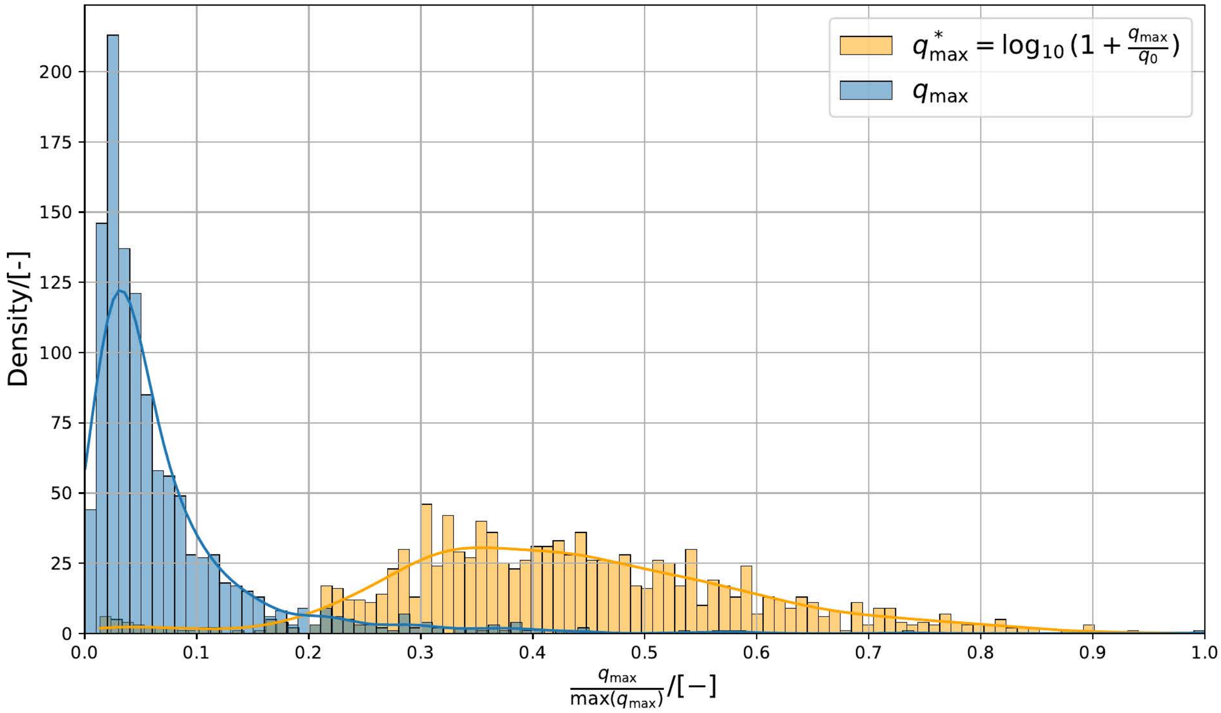


Figure 2: Probability distribution of the (non-zero) values of the maximum loading (non-transformed q_{max} in blue, transformed q_{max} in yellow). The blue distribution presents a right tail, which poses the risk of biasing. To deal with this, the yellow distribution is generated, resulting in a much better distributed data.

descriptors, such as the accessible surface area and volume of the zeolite, as well as the principal mass moments of inertia of the molecules.

Machine Learning Models

In a majority of the available literature, only a single ML model is used to predict the selected target. However, early attempts at using only a single model have resulted in overall commendable performance, but with many datapoints that are incorrectly predicted to be cases of positive adsorption. To this end, the proposed model is a cascading ML model, i.e. a model made out of two ML models placed in series. The motivation behind this choice is physical. The first model tries to predict whether adsorption is likely to occur as a binary outcome. If it is, the regressor will predict the actual amount. Otherwise, the value can be set to 0. This means that the classifier is trained in both datapoints of zero adsorption and non-zero adsorption, whilst the regressor is trained on non-zero adsorption datapoints only. Different models are considered for the second step. These include (extreme) gradient-boosted trees, Support Vector Machines and the TABPFN model. As all of these models are already discussed in detail in literature, only a short summary is provided here.

Gradient-boosted trees (GBT)[33,34] are an ensemble of decision trees that are trained sequentially. Since singular decision tree runs the risk of overfitting data, multiple trees are created. It works by making an ensemble of decision trees, where an initial tree is trained, and each tree made after this one tries to correct the residuals of the previous one. When constructing the next tree in the ensemble, pseudo-residuals defined as the negative derivative of a specified loss function evaluated on the current one is determined. A tree regressor is fitted on top of these pseudo-residuals, and is then added to the ensemble. The final model prediction is thus a combination of the initial weak performer added with contributions from each of the trees trained on pseudo-residuals. A variant of this model is the extreme gradient-boosted tree (XGB).[36] Traditional gradient-boosted trees only consider the first derivative of the loss function. To strengthen the boosting, XGB determines the next tree in the ensemble by minimizing an objective using not only first and second derivatives of the original loss function, but also a term penalizing the pseudo-residual tree complexity (via L2 regularization). Support Vector Regressors (SVR)[33,35] work by optimizing a function across a multi-dimensional space that does note deviate more than a specified ε margin, whilst minimizing model complexity. To allow these models to not compute the mapping directly, Support Vector Regressors use kernel functions to represent the non-linear relationships. Unlike the GBT and XGB methods, it is sensitive to data scaling due to its distance-based method. TABPFN is a Prior-Data Fitted Network trained over millions of synthetic tabular tasks, over which it learns an approximate to Bayesian inference. For regression tasks, it outputs a point estimate of the value from a predicted probability distribution of the target. This means that the training data is merely used to allow the model to condition the right prior. Because it is a pre-trained model, it does not need any hyperparameter tuning. Some of these models are subjected to a set of hyperparameters which determine its training dynamics, whose combinations can result in a better or worse performance. To find an optimal set of hyperparameters, Bayesian optimisation[58] is used. It aims to maximize a given objective by means of a probabilistic surrogate and an acquisition function guiding the optimizer. During each iteration, a value of the objective function is sampled at a certain point, which helps in actively updating a map of the optimization space by feeding it to the surrogate. In parallel, an acquisition function proposes a point which would result in the largest expected improvement, where the objective is resampled. This method is a common approach to optimizing model hyperparameters,[59] as it offers a more informed optimization than random or grid search, accuracy comparable to more complex methods, whilst still being computationally light. For this work, both the optimization of the classifier and regressor have are given 150 iterations with 10% dedicated to build the initial surrogate model. In this work, all models are implemented using Python. Both the GBT and SVM are implemented using scikit-learn,[60] XGB is implemented using the XGBoost,[36] and TABPFN is implemented through the TabPFN package.

Analysis Metrics

To quantitatively assess model performance, several performance metrics need to be established.[33] Since the cascade is a combination of a classifier and a regressor, different set of metrics need to be defined. Classifier performance is usually assessed using the accuracy. Given a set of outcomes made up of true positives (TP), true negatives (TN), false positives (FP) false negatives (FN), the accuracy is defined as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

This metric could be misleading if there is class imbalance in the data, which may not reflect poor prediction performance on the minority class. For this purpose, the class-wise $f1$-score is introduced. It represents how reliably a model can identify a class without missing real cases or raise false alarms. If is defined as

$$f1 = 2 \times \frac{P \times R}{P + R} \tag{7}$$

with precision P and recall R, both given as

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \tag{8}$$

It allows to both minimize the amount of false positives (via precision) and maximize capture of true positives (via recall). By averaging the score between classes, a macro-f_1

$$\text{MACRO-}f1 = \frac{1}{N} \sum_{c=1}^{N} f1_c \tag{9}$$

is created, ensuring that the final metric that the dominant and minority classes are fairly represented. This advantage makes the macro-$f1$ score a suitable choice as objective function for Bayesian optimization of the classifier. Performance of the regressor is primarily assessed using the coefficient of determination R^2, the mean absolute error MAE and the mean squared error MSE. Given a set of true values y with mean $\bar{y}$ and a set of predictions $\hat{y}$, both having N samples, these metrics are defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \tag{10}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \bar{y}| \tag{11}$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{12}$$

The mean square error is used as objective function for optimizing the regressor as differences are punished more severely while keeping the objective differentiable. To ensure a robust and unbiased estimate, the selected objectives are the result of a 5-fold cross-validation on the training set. This procedure splits the training set into 5 subsets (”folds”). The model is trained on four folds, and tested on the fifth one. This is repeated so that each fold is used as a test once, and the average across all folds is reported.

3. Results

Analysis of finite size effects analysis

A closer look is taken at the simulation results with RASPA2 to investigate to which extent finite size effects are present. Two typical examples are presented in Fig. 3. The top figures show the origin of finite-size effects and how the final result of the maximum loading is in-

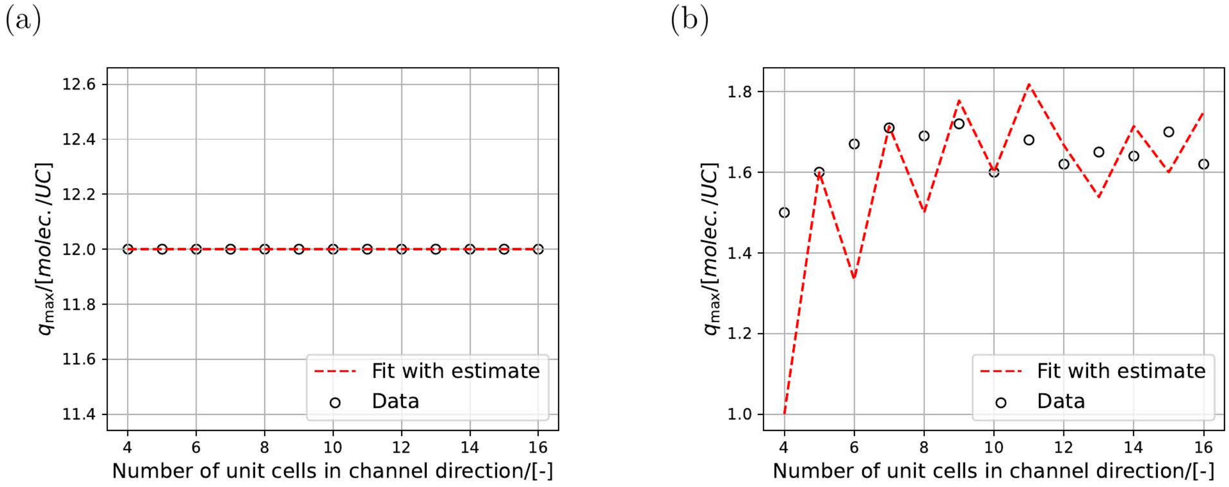


Figure 3: (a) Finite-size effects for adsorption of ethane in AET-type zeolite. Because ethane is much smaller than the channels of AET-type zeolite, the maximum loading changes only slightly as the simulation box increases. (b) Finite-size effects for adsorption of 2-methyl-4-ethyl-nonane in SAF-type zeolite. Larger molecules in narrower channels show stronger variations before converging (here around $\sim$ 1.65 molec./UC). The red line shows a fit to in Eq. 13

.

fluenced. Fig. 3a shows the finite-size effects of ethane in AET-type zeolite. Since ethane is small compared to the channel size, there is no variability in the final result as more unit cells are added. In sharp contrast, Fig. 3b, which shows 2-methyl-4-ethyl-nonane in SAF-type zeolite, demonstrates that the smallest simulation box does not produce a representative result. Because of the larger alkane and the narrower channel, more unit cells are required to get a better ensemble average, as demonstrated by the stabilization of the maximum loading value with increasing number of unit cells.

If one would use the principle behind Fig. 1, one could be compelled to fit a function to see how many unit cells in the channel direction are needed to obtain the result of an infinitely large domain. This can be approximated using the unit cell channel length L, the amount of unit cells in the channel direction x, and a fitted parameter a resembling the size of the molecule. The resulting function would be

$$E(a, L, x) = \frac{\mathrm{int}(xL/a)}{x} \tag{13}$$

where the function $\mathrm{int}(x)$ rounds its argument to the lowest integer. The fit is applied to the data of Fig. 3a and 3b, shown by the dashed red line. In Fig. 3b, it can be seen that the general trend is reproduced, but the nonlinearity introduced by the numerator of Eq. 13 leads to visible instability in the result. This illustrates the complexity of the problem at hand, as well as proves that simple function-fitting is insufficient to predict maximum loading of hydrocarbons in zeolites. This supports the use of machine learning for this specific task.

Correlation Analysis of Descriptors

To confirm the choice of descriptors, a correlation matrix can be constructed. Using a correlation coefficient, one can observe whether the feature has an impact on the maximum loading, or whether it is correlated to other features, thus making it redundant. Because of the non-linear nature of adsorption, the Spearman ρ_{S} coefficient[61] is used for this task as it not only can handle monotonic non-linear correlations, but can also handle outliers much better. A correlation between the descriptors themselves, as well as with the maximum loading is presented in Fig. 4. Only points with non-zero adsorption are considered for these correlations. Selecting descriptors that are correlated to the target helps the model capture clear relevant patterns, possibly enhancing the performance. However, correlated descriptors introduce redundancy, due to the overlapping information they provide. The latter may negatively impact the model because of multicollinearity, whilst also making the model more complex by addition of a new dimension. The first cluster that can be noticed is the one of zeolite descriptors on the top-left (ϕ, ASA, AVA, $AccV$, FD, $TD10$, d, RPD, LCD, ρ). From there, it can be noted that volume and topology-based descriptors are moderately correlated to q_{max}. This means that despite not solely determining q_{max}, these descriptors still impose some level of constraint. The void fraction, accessible area, accessible volumes, framework density, topological density, and gravimetric density highly correlate ($\rho_S > 0.9$). This is expected as the descriptors are mutually dependent on each other. These descriptors also moderately correlate to the maximum loading as larger channels increases volumes and areas, while driving densities down. It is important to state the absent correlation between RPD and q_{max}, which is expected as the steric hindrance of molecules in the channels is checked by the classifier.

Molecule descriptors exhibit stronger correlations to q_{max}, as well as between each other due to all descriptors being directly related to the size and shape of the molecule. Shape driven descriptors (such at the principal moments of inertia and the acentric factor) are seen to be highly correlated to the maximum loading, hinting that shape-based selectivity may be inherently present in the data. Correlations involving branching of the alkanes are seen to be low-to-moderate, and as such are expected to have limited impact. This is mostly because descriptors pertaining to the size of the molecules already include this information

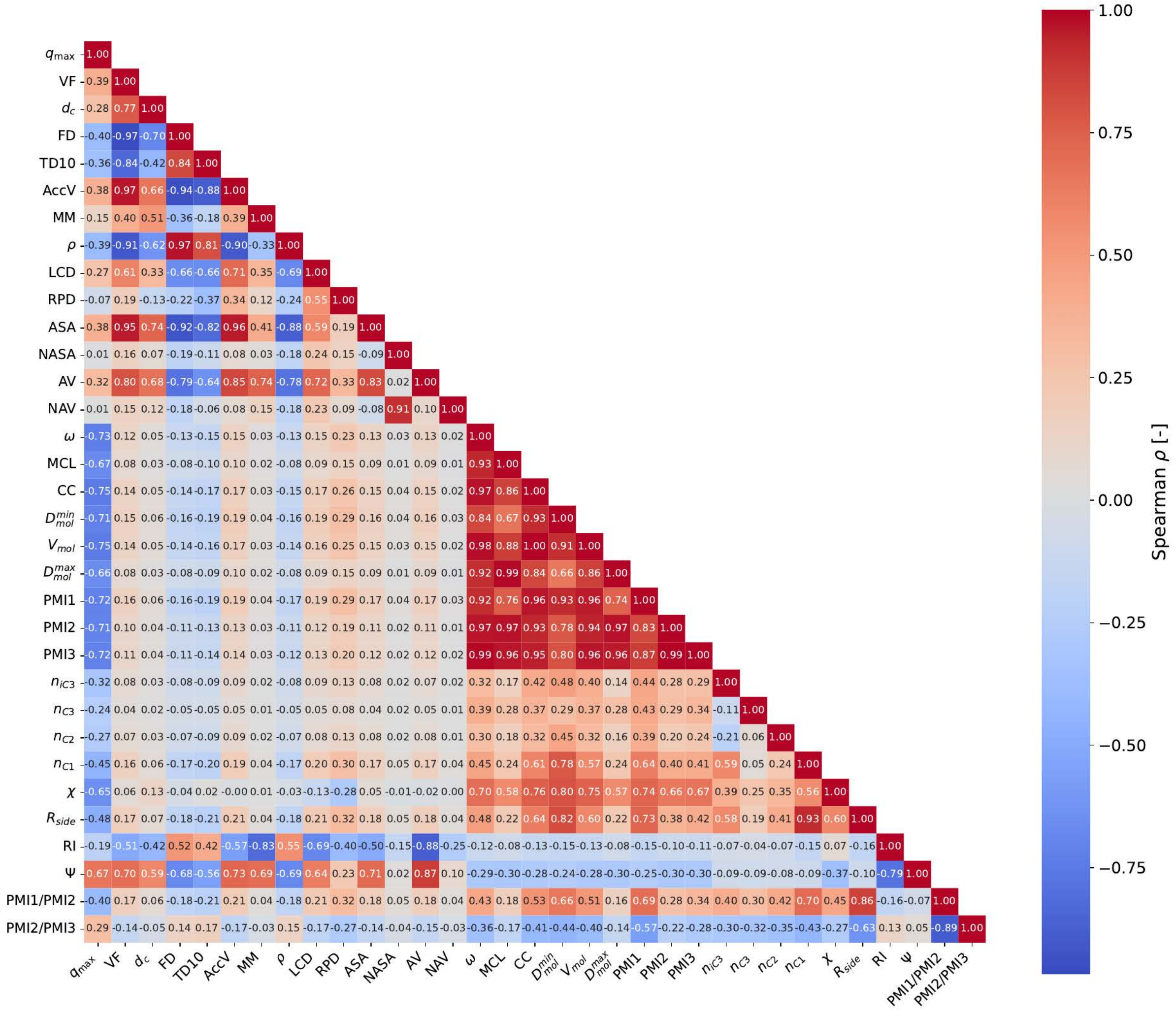


Figure 4: Spearman Correlation Matrix between descriptors and $q_{\max}$. Despite the fact that molecule and zeolite descriptors exhibit high internal correlation between each other due to their shared origin, it is clear that both are essential to predicting the maximum loading. Molecular descriptors appear to provide the dominant contribution, while zeolite-based descriptors add complementary information. All descriptors in this matrix are defined in table 1.

implicitly. Overall, it can be concluded from the matrix that the extent of the maximum loading is dependent on both zeolite and molecule descriptors, albeit there seems to already be a dominance of the latter in the final result. This is expected as these are known samples that get adsorbed, so there is no ”exclusion regime” that needs to be defined with pore geometry, and as such will operate more as soft upper bound. This allows molecular variation is expected to dominate here. It is also observed that a lot of descriptors are correlated with each other. This in itself is not fatal, but indicates that some of these descriptors may be removed if found to be of little importance to the model prediction.

Model Performance and Outlier Analysis

Different metrics are relevant to assess the performance of each stage of the cascade model. As such, each element of the cascade is assessed individually. The classifier performance is quantified by means of a confusion matrix and a performance table. Regressor performance is addressed by means of parity plots. All results were obtained using a fixed random seed of 57 and with 20% of the data withheld for testing to obtain consistent results.

Table 2: Summary table of classifier performance metrics (Eqs. 6, 7, 8, 9) for predicting adsorption and non-adsorption cases, both overall and per-class basis. The high accuracy demonstrates the model is reliably predicting the correct adsorption case. Class-wise precision, recall and $f1$-scores show that the model is more likely to predict a no-adsorption when adsorption should occur rather than vice-versa.

Class	Metric	Value
Overall	Accuracy	0.9938
$q_{max} = 0$	Precision	0.9929
	Recall	0.9982
	$f1$-score	0.9955
$q_{max} > 0$	Precision	0.9959
	Recall	0.9840
	$f1$-score	0.9899

The classifier confusion matrix and performance table obtained from the classifier can be found in Fig. 5 and table 2 respectively. As can be seen by the confusion matrix, the classifier has excellent performance, with a near perfect accuracy. This can be attributed to Bayesian optimisation finding an optimal set of training hyperparameters. It can be observed that from table 2 on a per-class basis, both have very high precision and recall. Despite this,

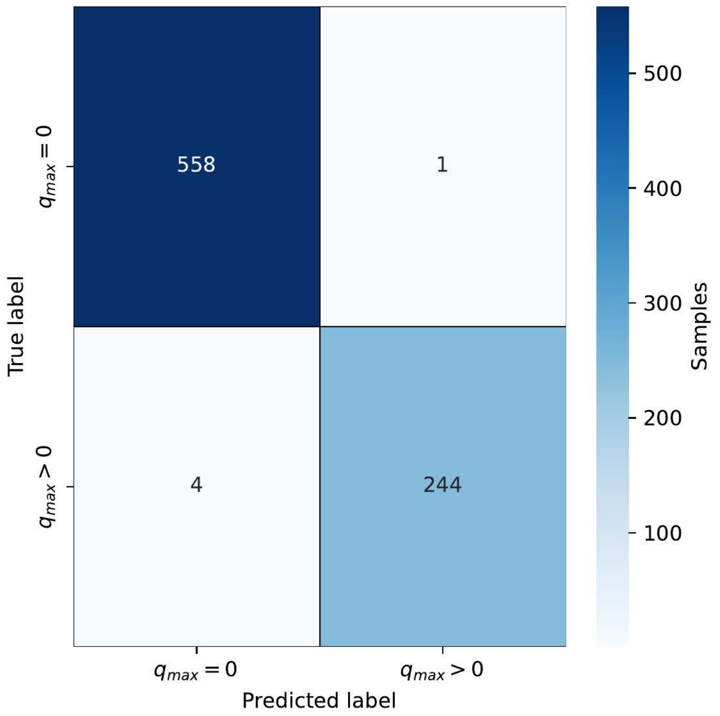


Figure 5: Gradient-boosted classifier confusion matrix. Each cell represents the number of samples. The low number of total mislabelled points reflects near-perfect accuracy. The model is slightly more prone to predicting zero adsorption for combinations that would adsorb, rather than incorrectly predicting adsorption where none should occur. These values are then used for Eqs. 6, 7, and 9.

the former is higher for adsorption cases, and the latter is higher for non-adsorption cases. This suggest that the classifier is more conservative, missing true adsorption cases (lower recall) rather that predicting adsorption happens when it should not happen (lower precision). This theory is further supported by the results in the confusion matrix. It can also be observed that the $f1$-scores of both classes are very close to each other, hinting that the model can predict both adsorption and no-adsorption cases with comparable strength. As shown in Fig. 5, only five samples were incorrectly labelled. These are listed in table 3, along side their true and predicted labels. The table above shows that there is not one molecule or zeolite particularly prone to being mislabelled. Closer analysis of these outliers reveals that UWY/3445-tetramethyl-4-isopropyloctane is missed by the classifier because it is the adsorbing combination with the highest value of the minimum molecular diameter-to-

Table 3: Incorrectly predicted alkane–zeolite combinations (test set)

Zeolite	Molecule	Predicted label	True label
NES	n-butane	$q_{\max} > 0$	$q_{\max} = 0$
AFO	n-nonane	$q_{\max} = 0$	$q_{\max} > 0$
MRE	2-methylpentane	$q_{\max} = 0$	$q_{\max} > 0$
UWY	3445-tetramethyl-4-isopropyloctane	$q_{\max} = 0$	$q_{\max} > 0$
VET	226-trimethyl-3-ethyloctane	$q_{\max} = 0$	$q_{\max} > 0$

restricting pore diameter ratio χ (~ 1.55). Since the combination with the second highest value, BEC/3445-tetramethyl-4-isopropyloctane is correctly labelled, suggesting that there is a threshold value of χ in the model beyond which adsorption is predicted as not happening. This is grounded in physics as the diameter-to-restricting pore diameter ratio accounts for the steric hindrance of molecules in the zeolite channels. A similar explanation applies to VET/226-trimethyl-3-ethyloctane and AFO/n-nonane, which both have the lowest values of the packing factor Ψ among adsorbing cases, thus also hinting to the presence of a volume-based threshold. As such, a possible strategy to improve the model would be to generate more training data close to these islands of data.

For the regressor stage, each of the models is trained and fitted to the maximum loadings obtained from RASPA2. The parity plots for each of the regressors are presented in the figure below. This includes their respective values of R^2, MAE and MSE. The resulting plots can be seen on a pre-regressor basis in Fig. 6. The ideal fit line ($q_{\max}$(predicted) = $q_{\max}$(actual)) is depicted in a red dashed line in all plots. One thing that is present in all models is a patch of slight over-prediction at the lower left, i.e. the region of low maximum loadings. This regime is mostly governed by large and highly branched molecules. Since the metrics get better with the models, this means that what changes is the quality of prediction on the bulk range. This can be noticed by the spread around the points getting closer to the ideal line as the model changes.

Out of all models, TABPFN performs the best with the highest R^2 and lowest errors (both MAE and MSE). This is followed by XGB, GBT, and the SVM. The SVM achieves decent accuracy but is known to struggle with complex nonlinear dependencies and high-dimensional

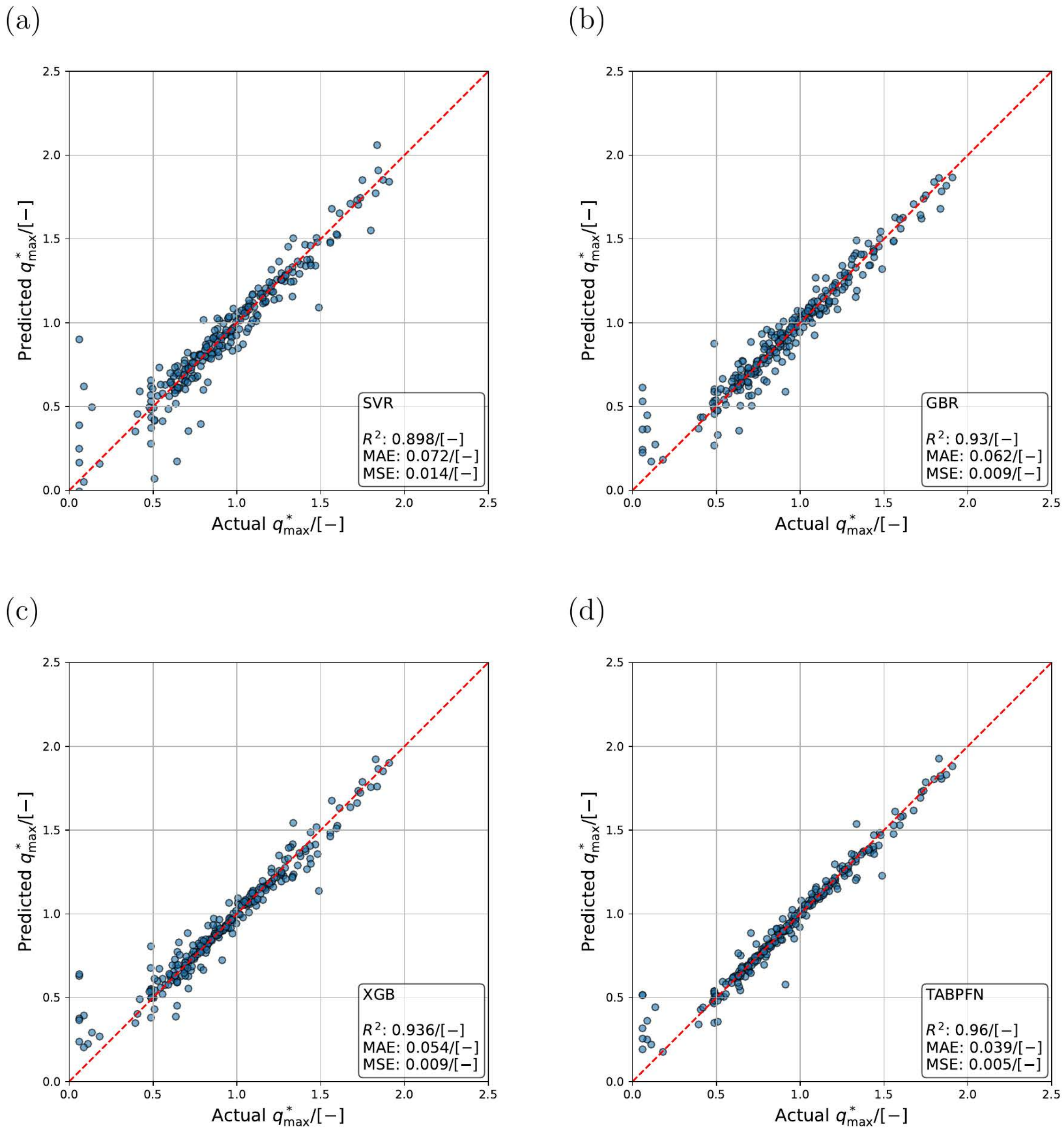


Figure 6: Parity plots for all four considered regressor models with their associated performance metrics. All models exhibit high performance, though the very low maximum loading regime dominated by larger and more bulky molecules remains challenging. SVM lags behind mostly due to the high dimensionality of the problem. GBR and XGB perform well, while TabPFN achieves the best performance by capturing dependencies that the other models cannot.

feature interactions, leading to systematic under-prediction for larger or geometrically intricate molecules. Since the tree-based models are over-performing by some margin, it could also be sign that the decision boundary the model needed to fit had overlap zones, which is a known limitation for SVRs.[62] The tree-based methods (GBR and XGB), due to their boosting capabilities, can use the residuals to create better trees. This explains the increased performance, with XGB being slightly better thank to built in mechanisms that the GBR does not have (e.g. regularization and depth-first pruning). Lastly, TABPFN exhibits the best metrics thanks to its transformer-based architecture, allowing it to capture more complex relationships that would otherwise not be captured.

Model Interpretability

ML models are considered as black boxes, as it is hard to understand the decisions made by a model solely by looking at its parameters.[63] This can be detrimental to reliability and scientific relevance of the model, as it may not confirm whether the model learned the intended pattern. SHapley Additive exPlanation (SHAP) analysis can help confirm this.[64] Based on the Shapley values from game theory,[63] this method offers a way to quantify how much each feature contributes to the final prediction of the model. It is important to note that the actual values themselves carry no meaning, as only the sign of the resulting values (whether it pushes the prediction up or down) is of interest. The $SHAP$ value distributions of both classifier and regressor models can are presented in Fig. 7 respectively. These results are both obtained by fixing the random seed at 78 and using the GBT regressor as it is the highest accuracy model that could work with SHAP. It is clear by looking at Fig.7a that the classifier relies mostly on steric constraints when predicting the likelihood of adsorption. This is shown by the ratio between minimum molecular diameter and zeolite restricting pore diameter χ being the most influential feature. High values of χ (red) are clustering in the negative SHAP region, which shows the systematic negative impact of molecules that are too large to fit in the channel. This shows that not only did the descriptor work as intended, but also that the model itself is physically informed. This is further confirmed by RPD being the second most prominent descriptor with larger values pushing the prediction towards adsorption likelihood. The parameters Ψ and AV are also shown to also have major impact. While

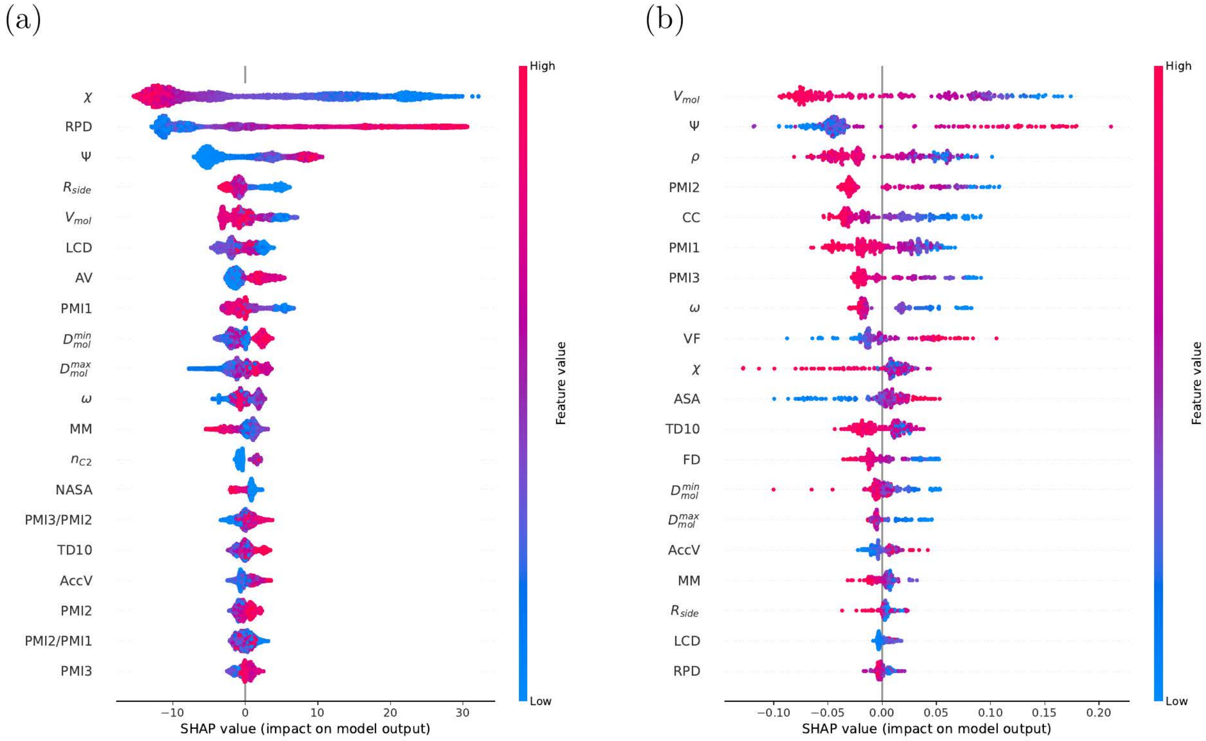


Figure 7: (a) SHAP value distribution of the classifier. χ and RPD dominate the outcome, showing that the model captures steric hindrance in the zeolite channels. Important molecule-based descriptors include PMI1 and R_{side}, indicating the model accounts for diffusion limitations of bulkier molecules. (b) SHAP value distribution of the regressor. Volume-based constraints dominate predictions via V_{mol}, Ψ, and ρ, with contributions from molecular shape descriptors such as principal moments of inertia, ω, and $D_{\text{mol}}^{\text{min}}/D_{\text{mol}}^{\text{max}}$, as well as R_{side}.

secondary to χ and RPD, these descriptors have major influence as these tell how much the molecule can move once inside the pore network. Certain molecule-based descriptors such as the first principal mass moment of inertia $PMI1$ and side-to-total carbon count ratio R_{side} also influence the prediction, as each serves as a proxy to the molecule bulkiness. However, it is smaller than the zeolite-based descriptors as steric constraints imposed by the channels dominate the adsorption process. The remaining descriptors follow the same logic of either representing the steric constraint imposed by the channel (such as LCD, AV, $NASA$, $TD10$ and $AccV$) or the overall size and bulkiness of the molecule (such as $D_{\text{mol}}^{\text{min}}$, $D_{\text{mol}}^{\text{max}}$, ω, n_{C2}, $\frac{\text{PMI1}}{\text{PMI2}}$ and $\frac{\text{PMI2}}{\text{PMI3}}$). Another observation is that as descriptors become less important (and as distributions narrow down), the influence becomes less monotonic and interpretable. This hints that the most important descriptors are used early on in the prediction process and impose clear-cut boundaries. Less important descriptors are thus used for more localized splits of data.

The SHAP distribution for the regressor featured in Fig.7b shows that the leading descriptors are the molecule volume V_{mol} and the ratio between the available zeolite volume and molecule volume Ψ, meaning that the fundamental aspect behind the prediction is the volume available for a given molecule. This constitutes another steric constraint encapsulated by Ψ that proves that the regressor as well is physics-informed. Notably, Ψ shows a context dependence: more compact alkanes in larger channels tend to have higher maximum loadings. A noteworthy element is the presence of all three principal mass moments of inertia, the acentric factor, both minimum and maximum diameters as well as R_{side}, meaning that the shape of the molecule also contributes to the final result. This proves shape-based selectivity is also taken into account to some extent. Indeed, the spreads indicate that larger, bulkier and less spherical molecules will also have lower maximum loadings. Structural constraints given by the accessible surface area, as well as topological and framework densities play a role, albeit minor compared to the previous two. A last remark that can be made is about how the amount of side groups and their nature are being used by the models. In the classifier, it can be observed that the side-to-main chain carbon count R_{side} has some influence over the final prediction, but other than the amount of ethane groups, no side-group counts are present. Moreover, the regressor stage completely disregards the amount of side-groups, but still uses the total carbon count and R_{side} for its prediction. This shows that when computing the maximum uptake, the nature of the side groups has little impact. This can be traced to the choice of descriptors: since molecular volume is already used, it implicitly accounts for the side groups. As such, the latter is used mostly in more localized downstream in the tree.

Model Robustness

A critical aspect of the model to assess is its robustness, i.e. its resilience against changes in the training dataset. In this work, three tests are carried out to assess how performance changes as essential training parameters are changed. Each component of the cascade is evaluated using a single representative metric: the classifier is assessed with macro $f1$-score and the regressor via R^2. For this test, TABPFN is used as regressor as it proved to be the best performer and interpretability is not an issue in this context. The raw data for the study are present in the Supporting Information.

A first robustness assessment revolves around changing the random seed used to split the dataset as well as initialize both the model and the Bayesian optimizer. This assesses how resilient the model is against initialization and random splits in the data. This can also give a reasonable idea on what the expected performance of the model would be across multiple runs, and whether the reported metrics are representative or overly optimistic. To carry out this test, 30 random seeds are generated via a random integer generator and fed to the model.

The box-and-whiskers plots on the left column show of Fig. 8 that despite an outlier on the regressor end (from random seed `210`), both models of the cascade present minimal variation. The random-seed average of the macro $f1$-score is noted to remain steady through all random seeds with an asymptotic value of $\langle \text{MACRO-}f_1 \rangle_\infty = 0.991$. This can be attributed to the Bayesian optimization step when training the model as this ensures that the best hyperparameters are used for the model. While the random seed average R^2 of the TABPFN regressor initially dips during early iterations, it quickly stabilizes to a final value of $\langle R^2 \rangle_\infty = 0.955$. This hints that although some data splits may lead to a more favourable performance on the test set, the overall expected performance is close to what was initially seen in Fig.6d. This helps conclude that the cascade is robust against data splits and optimization initialization.

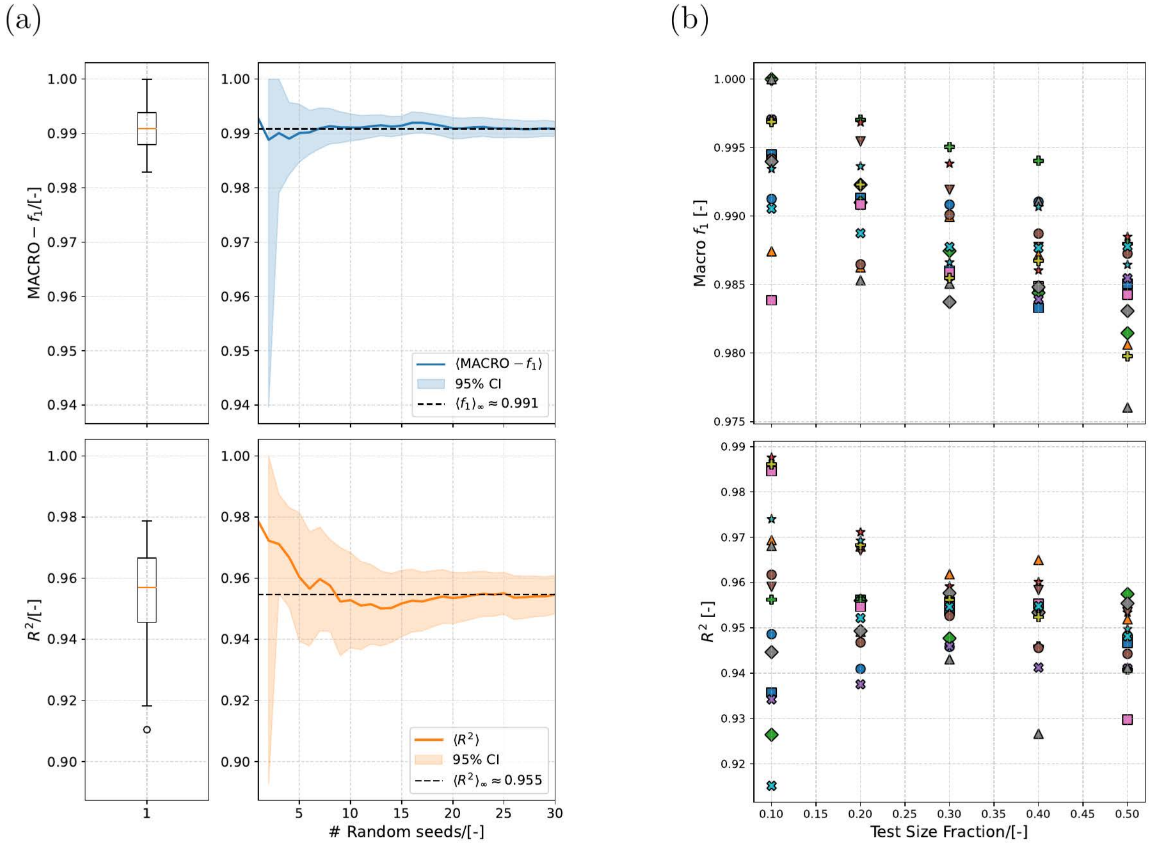


Figure 8: (a) Box and whisker plots as well as moving average across random seeds of both the classier MACRO − $f1$ (defined by Eq.9) and regressor R^2 (defined in Eq.10). The classifier performance is not affected by the changes in random seed, remaining above 0.99 for the MACRO − $f1$. TABPFN dips slightly initially but stabilizes after a few iterations, showing the robustness of both elements of the cascade. (b) Values of the classier MACRO − $f1$ and regressor R^2 across multiple random seeds. Despite the GBT classifier decreasing in performance as test size gets larger, the performance is still very high. However, the performance of the regressor remains acceptable, but gets less consistent as test set size is made smaller, which may indicate overfitting.

A second assessment changes the size of the testing set. Doing so provides insight in whether the performance of the model is affected by the availability of training data. To get a more representative result, the average result across 15 random seeds is used. The results are visible in Fig.8b. Split sizes range from 10% to 50% of all data available at increments of 10%. Despite the decrease in the macro $f1$-score as the test set grows, performance is kept high. This indicates that the classifier itself is robust against dataset sizes. Furthermore, the low variance in values across the random seeds further confirm the insights from Fig.8a showing that randomness has very limited effect in performance. The regressor is observed to have more variance across results, with the largest variance for a test dataset size of 10% or all data. Since the largest variance across the results are noted for the smaller test size fractions across random seeds, this hints that the model may be overfitting the data, where the training data is very well fitted but performance on the test set is not maintained. Furthemore, TABPFN being a transformer based model, it is more inclined to memorize patterns rather than generalize.

Machine Learning versus Deep Learning

The cascade model used previously is made out of classical ML models. Increasing focus has been shifted towards using Neural Networks as one can use inputs that would not be very hard to code for regular ML, albeit to the cost of lower interpretability. To this end, an Artificial Neural Network (ANN) is created as a benchmark to see whether it can provide an alternative the current cascade model. The artificial neural network used in this study is implemented using PyTorch, and is a fully-connected feed-forward network. The architecture of the network constitutes of an input layer with the same dimensionality as the number of descriptor, two hidden layers with 8 neurons each, as well as an output layer with one neuron that predicts the maximum loading as a continuous variable. Each of the hidden layers employ ReLu activation functions. The model is trained using the Adam optimiser with a learning rate of 0.001 [-] and MSE as a loss function. Training is performed across 10000 epochs. Despite the possibility to use CUDA toolkit on the ANN to accelerate training, it was decided to use CPU-based training to keep the comparison fair since scikit-learn and XGBoost have very limited capability to move training to the GPU. Training was done on a Dell Precision 5690 laptop with an Intel Core Ultra 7 165H processor.

In this context, the deployability of such an alternative (i.e. the amount of time and computing power needed to train the model) when compared to the current GBT classifier+TABPFN cascade is investigated. To do so, the total training time, per-sample inference time and model size are considered. The results are presented in Fig. 4. It is to be noted that the time spent on Bayesian optimisation is omitted as it is only done once and does not contribute towards model runtime performance.

Table 4: Cascade and ANN training time, per-sample inference time and model size

	Cascade model (GBC/TABPFN)	**ANN**
Training wall time [s]	0.6479/2.5151	18.8565
Training CPU time [s]	0.6406/0.7343	76.0625
Per-sample inference time [s/10^{-3}]	0.0024/6.3335	1.9377
Model size [MB]	0.1187/42.1429	0.0039
Used RAM [MB]	0.0078/248.2266	16.4531

Right away, it can be observed that the wall and CPU times for training the cascade are lower than for the ANN, thus indicating that the former is cheaper to train. This is expected since the neural network's training procedure involves an active step of optimization to find the ideal weights of the network, whilst GBC is fast to optimize and TabPFN is already pre-trained.

The values for the inference time are seen to vary substantially between the approaches. The GBC in the cascade model is proven to be very fast whilst TabPFN is much slower due to its transformer-based nature, and the ANN has an inference time between the two. However, it is to be noted that unlike the ANN, TabPFN only has to work for samples that have non-zero adsorption rather than on all samples. As such, the cascade outperforms the ANN. It is also visible that the cascade has a larger footprint in both RAM and disk memory. No matter the case, the values at this stage are still very modest, and as such do not. However, this does raise the possibility of using a neural network to substitute the two stage model.

4. Conclusions

In this work, an ML framework was developed to predict the maximum loading of alkane isomers in zeolites. The final model relies on the combination of a gradient-boosted tree for the classifier determining if adsorption occurs and a regressor stage predicting the actual maximum loading. Both take in data primarily relating to the molecule and zeolite size and volume, with several topological descriptors as well as some composite descriptors to enhance the interactions between the two. The classifier labels the cases with almost perfect performance, despite having a minor bias towards predicting no adsorption in case of doubt. Out of all considered regressors, the TabPFN model preformed best. An interpretability study has also been carried out using SHAP values. It was shown that both stages are providing predictions that are in line with the physics behind adsorption in zeolites, including steric hindrance and shape-based selectivity. A robustness study has been carried out on the GBT+TabPFN cascade, which showed that the full model is robust against random data splits, but the regressor performance varies a lot with too little testing data, indicating possible overfitting. Finally, the cascade model was compared to a neural network, which showed that despite a lower training time and cost, the cascade model is heavier on RAM and disk space due to the size of the TabPFN model itself. Future research on this topic may include addressing the problem of the low-value maximum uptake regime, either by altering the dataset (either by adding more samples to the the full dataset or by oversampling the problematic region when creating the training set) or by reviewing the cascade model (e.g. have two regressors handling smaller and larger molecules respectively). Another possible extension to this work may be the inclusion of descriptors relating to the channel connectivity, such as ones based in network science (e.g. degree or betweenness centrality). As data set grows larger and the descriptors more complex, it may be wise to consider transitioning to deep learning methods if the accuracy is shown to improve.

Acknowledgement

The author acknowledges the use of computational resources of the DelftBlue supercomputer, provided by the Delft High Performance Computing Center (https://www.tudelft.nl/dhpc), as well as the use of Snellius supercomputer, provided by SURF (https://www.surf.nl/diensten/rekenen/snellius-de-nationale-supercomputer).

Supporting Information Available

The Supporting Information contains the file SI1.pdf, which contains the list of considered zeolites, the list of considered alkanes, the relevant information as to the Bayesian Optimization and the raw results from the robustness study. SI2.xlsx contains an excel table with all RASPA2 simulation results. It also contains the actual dataset that is used to train all the models. The SI3 folder contains all the Python scripts used for dataset generation (`ML_GENDATASET.py`), training and optimizing all ML models (`ML_CHAINED_MODELS.py`), the robustness study (`ML_ROBUSTNESS.py`), training the Artificial Neural Network (`ML_NNARCHITECTURE.py`). It also contains the supporting functions used to create the data, including the script that collects the IZA data online (`SF_IZAONLINE.py`), the sampling method used to obtain the set of considered molecules (`SF_RANDMOL.py`) and the set used to generate all the data needed for the alkanes using RDKit (`SF_LCH.py`).

References

(1) United Nations Framework Convention on Climate Change (UNFCCC) Report of the Conference of the Parties on its Twenty-First Session, held in Paris from 30 November to 13 December 2015. `https://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf`, 2015; Accessed: 2025-12-11.

(2) Wong, F. W. M. H.; Kez, D. A.; Del Rio, D. F.; Foley, A.; Rooney, D.; Abai, M. Decarbonizing and offsetting emissions in the airline industry: Current perspectives and strategies. *Energy* **2024**, *313*, 133809.

(3) Song, G.; An, H.; Hou, Y.; Tong, H.; Liu, J.; Tang, X.; Yi, H. Review of the historical trends and decarbonization pathways of the civil aviation sector. *Renewable and Sustainable Energy Reviews* **2025**, *222*, 115927.

(4) van Dyk, S.; Saddler, J. *Progress in the Commercialization of Biojet / Sustainable Aviation Fuels (SAF): Technologies, potential and challenges*; Technical Report, 2021; Accessed: 2025-10-31.

(5) Calderon, O. R.; Tao, L.; Abdullah, Z.; Talmadge, M.; Milbrandt, A.; Smolinski, S.; Moriarty, K.; Bhatt, A.; Zhang, Y.; Ravi, V. et al. *Sustainable Aviation Fuel State-of-Industry Report: Hydroprocessed Esters and Fatty Acids Pathway*; 2024.

(6) Sharma, S.; Baur, R.; Rigutto, M.; Zuidema, E.; Agarwal, U.; Calero, S.; Dubbeldam, D.; Vlugt, T. J. H. Computing entropy for Long-Chain alkanes using Linear regression: Application to hydroisomerization. *Entropy* **2024**, *26*, 1120.

(7) Han, Y.; Yuan, J.; Xing, M.; Cao, J.; Chen, Z.; Zhang, L.; Tao, Z.; Liu, Z.; Zheng, A.; Wen, X. et al. Shape selectivity of zeolite for hydroisomerization of long-chain alkanes. *New Journal of Chemistry* **2023**, *47*, 1401–1412.

(8) Misra, P.; Alvarez-Majmutov, A.; Chen, J. Isomerization catalysts and technologies for biorefining: Opportunities for producing sustainable aviation fuels. *Fuel* **2023**, *351*, 128994.

(9) Smit, B.; Maesen, T. L. M. Molecular simulations of zeolites: adsorption, diffusion, and shape selectivity. *Chemical Reviews* **2008**, *108*, 4125–4184.

(10) Smit, B.; Frenkel, D. *Understanding Molecular Simulation*, 3rd ed.; Elsevier: 125 London Wall, London EC2Y 5AS, United Kingdom, 2023.

(11) Daou, A. S. S.; Fang, H.; Boulfelfel, S. E.; Ravikovitch, P. I.; Sholl, D. S. Machine Learning and IAST-Aided High-Throughput Screening of Cationic and Silica Zeolites for Alkane Capture, Storage, and Separations. *Journal of Physical Chemistry C* **2024**, *128*, 6089–6105.

(12) Hypothetical Zeolites Database. `http://www.hypotheticalzeolites.net/`, Accessed: 2025-12-11.

(13) Rieder, S. R.; Oliveira, M. P.; Riniker, S.; Hünberger, P. H. Development of an open-source software for isomer enumeration. *Journal of Cheminformatics* **2023**, *15*, 10.

(14) Sharma, S.; Yang, P.; Liu, Y.; Rossi, K.; Bai, P.; Rigutto, M. S.; Zuidema, E.; Agarwal, U.; Baur, R.; Calero, S. et al. Machine Learning-Based Predictions of Henry Coefficients for Long-Chain Alkanes in One-Dimensional Zeolites: Application to Hydroisomerization. *Journal of Physical Chemistry C* **2025**, *129*, 18234–18249.

(15) Langmuir, I. The adsorption of gases on plane surfaces of glass, mica and platinum. *Journal of the American Chemical Society* **1918**, *40*, 1361–1403.

(16) Myers, A. L.; Prausnitz, J. M. Thermodynamics of mixed-gas adsorption. *AIChE Journal* **1965**, *11*, 121–127.

(17) Radke, C. J.; Prausnitz, J. M. Thermodynamics of multi-solute adsorption from dilute liquid solutions. *AIChE Journal* **1972**, *18*, 761–768.

(18) Walton, K. S.; Sholl, D. S. Predicting multicomponent adsorption: 50 years of the ideal adsorbed solution theory. *AIChE Journal* **2015**, *61*, 2757–2762.

(19) Krishna, R.; Van Baten, J. M. How reliable is the ideal adsorbed solution theory for the estimation of mixture separation selectivities in microporous crystalline adsorbents? *ACS Omega* **2021**, *6*, 15499–15513.

(20) Makhanya, N. P.; Kumi, M.; Mbohwa, C.; Oboirien, B. Application of machine learning in adsorption energy storage using metal organic frameworks: A review. *Journal of Energy Storage* **2025**, *111*, 115363.

(21) Mai, H.; Le, T. C.; Chen, D.; Winkler, D. A.; Caruso, R. A. Machine Learning in the Development of Adsorbents for Clean Energy Application and Greenhouse Gas Capture. *Advanced Science* **2022**, *9*.

(22) Yang, C.; Qi, J.; Wang, A.; Zha, J.; Liu, C.; Yao, S. Application of machine learning in MOFs for gas adsorption and separation. *Materials Research Express* **2023**, *10*, 122001.

(23) Altintas, C.; Altundal, O. F.; Keskin, S.; Yildirim, R. Machine Learning Meets with Metal Organic Frameworks for Gas Storage and Separation. *Journal of Chemical Information and Modeling* **2021**, *61*, 2131–2146.

(24) Xue, X.; Cheng, M.; Wang, S.; Chen, S.; Zhou, L.; Liu, C.; Ji, X. High-Throughput Screening of Metal–Organic Frameworks Assisted by Machine Learning: Propane/Propylene Separation. *Industrial & Engineering Chemistry Research* **2023**, *62*, 1073–1084.

(25) Xiuying, L.; Chen, H.; Yuan, J.; Huang, J.; Li, X.; Yu, J. Revealing the struc-

ture–property relationship of all-silica zeolites for the carbon dioxide capture: a high throughput screening study. *Zeitschrift für Naturforschung A* **2023**, *78*, 863–873.

(26) Tatlier, M.; Munz, G.; Henninger, S. K. Relation of water adsorption capacities of zeolites with their structural properties. *Microporous and Mesoporous Materials* **2018**, *264*, 70–75.

(27) Evans, J. D.; Coudert, F.-X. Predicting the Mechanical Properties of Zeolite Frameworks by Machine Learning. *Chemistry of Materials* **2017**, *29*, 7833–7839.

(28) Yu, X.; Choi, S.; Tang, D.; Medford, A. J.; Sholl, D. S. Efficient Models for Predicting Temperature-Dependent Henry's Constants and Adsorption Selectivities for Diverse Collections of Molecules in Metal–Organic Frameworks. *Journal of Physical Chemistry C* **2021**, *125*, 18046–18057.

(29) Li, W.; Li, W.; Busch, A.; Wang, L.; Anggara, F.; Yang, S. Machine Learning Algorithm to Predict Methane Adsorption Capacity of Coal. *Energy & Fuels* **2024**, *38*, 23422–23432.

(30) Zhao, L.; Zhang, Q.; He, C.; Chen, Q.; Zhang, B. J. Quantitative Structure–Property Relationship Analysis for the Prediction of Propylene Adsorption Capacity in Pure Silicon Zeolites at Various Pressure Levels. *ACS Omega* **2022**, *7*, 33895–33907.

(31) Chakraborty, A.; Gandhi, A.; Hasan, M. F.; Venkatasubramanian, V. Discovering zeolite adsorption isotherms: a hybrid AI modeling approach. *Computer Aided Chemical Engineering* **2024**, *53*, 511–516.

(32) Chakraborty, A.; Gandhi, A.; Hasan, M. F.; Venkatasubramanian, V. Explainable AI modeling of zeolite adsorption isotherms. *Chemical Engineering Science* **2026**, *320*, 122361.

(33) Guido, S.; Mueller, A. C. *Introduction to machine learning with python*; O'Reilly Media, 2016; p 392.

(34) Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* **2001**, *29*, 1189–1232.

(35) Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer New York, 2009.

(36) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **2016**, 785–794, Also available as arXiv:1603.02754.

(37) Hollmann, N.; Müller, S.; Eggensperger, K.; Hutter, F. TabPFN: A Transformer that Solves Small Tabular Classification Problems in a Second. Proceedings of the International Conference on Learning Representations (ICLR). 2023.

(38) Dubbeldam, D.; Calero, S.; Ellis, D. E.; Snurr, R. Q. RASPA: molecular simulation software for adsorption and diffusion in flexible nanoporous materials. *Molecular Simulation* **2016**, *42*, 81–101.

(39) Dubbeldam, D.; Torres-Knoop, A.; Walton, K. S. On the inner workings of Monte Carlo codes. *Molecular Simulation* **2013**, *39*, 1253–1292.

(40) Bai, P.; Tsapatsis, M.; Siepmann, J. I. TrAPPE-ZEO: Transferable potentials for phase equilibria force field for All-Silica Zeolites. *Journal of Physical Chemistry C* **2013**, *117*, 24375–24387.

(41) Liu, B.; Smit, B.; Calero, S. Evaluation of a new force field for describing the adsorption behavior of alkanes in various pure silica zeolites. *Journal of Physical Chemistry B* **2006**, *110*, 20166–20171.

(42) Bingel, L. W.; Chen, A.; Agrawal, M.; Sholl, D. S. Experimentally Verified Alcohol Adsorption Isotherms in Nanoporous Materials from Literature Meta-Analysis. *Journal of Chemical Engineering Data* **2020**, *65*, 4970–4979.

(43) Gómez-Carracedo, M.; others Multivariate prediction of eight kerosene properties. *Fuel* **2003**, *82*, 1913–1921.

(44) Bajusz, D.; Rácz, A.; Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics* **2015**, *7*, 20.

(45) Baerlocher, C.; McCusker, L. B.; Brouwer, D.; Marler, B. Database of Zeolite Structures. `https://www.iza-structure.org/databases/`, 2025; Accessed: 2025-09-01.

(46) Dubbeldam, D.; Calero, S.; Vlugt, T. J. H. iRASPA: GPU-accelerated visualization software for materials scientists. *Molecular Simulation* **2018**, *44*, 653–676.

(47) Baerlocher, C.; McCusker, L. B.; Olson, D. H. *Atlas of Zeolite Framework Types*, 6th ed.; Elsevier, 2007; pp 3–11.

(48) Willems, T. F.; Rycroft, C. H.; Kazi, M.; Meza, J. C.; Haranczyk, M. Algorithms and tools for high-throughput geometry-based analysis of crystalline porous materials. *Microporous and Mesoporous Materials* **2012**, *149*, 134–141.

(49) Landrum, G. RDKit: Open-source cheminformatics. 2024; `https://www.rdkit.org`, Accessed: 2024-09-01.

(50) Vlugt, T. J. H.; Schenk, M. Influence of framework flexibility on the adsorption properties of hydrocarbons in the zeolite silicalite. *Journal of Physical Chemistry B* **2002**, *106*, 12757–12763.

(51) Krishna, R.; Smit, B.; Calero, S. Entropy effects during sorption of alkanes in zeolites. *Chemical Society Reviews* **2002**, *31*, 185–194.

(52) Sharma, S.; Rigutto, M. S.; Zuidema, E.; Agarwal, U.; Baur, R.; Dubbeldam, D.; Vlugt, T. J. H. Understanding shape selectivity effects of hydroisomerization using a reaction equilibrium model. *Journal of Chemical Physics* **2024**, *160*.

(53) Schenk, M.; Calero, S.; Maesen, T. L. M.; Vlugt, T. J. H.; van Benthem, L. L.; Verbeek, M. G.; Schnell, B.; Smit, B. Shape selectivity through entropy. *Journal of Catalysis* **2003**, *214*, 88–99.

(54) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society* **1992**, *114*, 10024–10035.

(55) Tarzia, A. mol-ellipsize: Molecular size calculation based on ellipsoid fitting over conformer ensembles. Software, version 1.0.1, available at `https://github.com/andrewtarzia/mol-ellipsize`, 2021; Accessed: 2025-10-31.

(56) James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An introduction to statistical learning*; 2013.

(57) West, R. M. Best practice in statistics: The use of log transformation. *Annals of Clinical Biochemistry International Journal of Laboratory Medicine* **2021**, *59*, 162–165.

(58) Wu, J.; Tian, Z.; Guo, W.; Yang, L. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology* **2019**, *17*, 26–40.

(59) Liu, C.; Balasubramanian, P.; An, J.; Li, F. Machine learning prediction of ammonia nitrogen adsorption on biochar with model evaluation and optimization. *npj Clean Water* **2025**, *8*, 13.

(60) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blon-

del, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(61) Spearman, C. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* **1904**, *15*, 72.

(62) Proceeding Book of the y-BIS Conference 2019: Recent Advances in Data Science and Business Analytics. 2019.

(63) Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 1st ed.; Leanpub: Online, 2020; Online book, accessed on 2025-01-31; Licensed under CC BY-NC-SA 4.0.

(64) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017; pp 4765–4774, To appear in NeurIPS 2017; arXiv preprint arXiv:1705.07874.

Supporting Information: Predicting the Maximum Loading in Zeolites for Hydroisomerization Applications: A Machine Learning Approach

Eric Johnsson,[†] Shrinjay Sharma,[‡] Arvind Gangoli Rao,[†] David Dubbeldam,[¶] Sofia Calero,[§] and Thijs J. H. Vlugt[*,‡]

[†] *Flight Performance & Propulsion Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands*

[‡] *Process & Energy Department, Faculty of Mechanical Engineering, Delft University of Technology, Leeghwaterstraat 39, 2628CB Delft, The Netherlands*

[¶] *Van 't Hoff Institute of Molecular Sciences, University of Amsterdam, Science Park 904, 1098XH, Amsterdam, The Netherlands*

[§] *Department of Applied Physics, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands*

E-mail: t.j.h.vlugt@tudelft.nl

List of considered zeolite structures

The table below represents all zeolite structures considered in this work. From the 261 recognized structures by the IZA,[1] these are the ones that: (1) have orthorombic unit cells; (2) have unit cells large enough to make computations fast; (3) have few to no enclosed voids, and; (4) have an accessible channel network.

Table S1: Zeolite structures considered in this work.

Channel dimensionality	Structure codes
1	AEL, AET, AFO, ATN, ATN, ATS, ATV, AWW, BOF, CAS CHI, DON, ESV, EZT, GON, IFO, JRY, MRE, MTF, PAR PON, PSI, RON, RWR, SAF, SAS, SFH, SSY, TON, VET
2	AEN, AFR, APC, APD, AWO, CDO, EON, ETL, FER, IHW ITE, IWV, MFS, MOR, NES, SEW, SFG, STI, STI, TER UEI, UFI, ZON
3	AEI, BEA, BEC, BOG, BSV, CGS, CLO, GOO, IFU, IMF ISV, ITH, ITR, ITV, IWR, IWS, IWW, JOZ, JSR, JST KFI, MEL, MER, MFI, MSE, MWF, NAT, NPT, OBW, PAU PHI, POS, PUN, RHO, RWY, SAO, SAV, SBE, SFV, SIV SZR, TSC, OUV, UWY, VSV

List of considered alkanes (linear and branched isomers)

The tables below constitute a list of all 47 alkanes used in training and testing the model. New alkanes are selected based on similarity to previous ones to obtain the most unique set of alkanes possible.

Table S2: Alkanes considered.

Name	SMILES
Methane	C
Ethane	CC
Propane	CCC
Butane	CCCC
Pentane	CCCCC
Hexane	CCCCCC
Heptane	CCCCCCC
Octane	CCCCCCCC
Nonane	CCCCCCCCC
23-dimethylbutane	CC(C)(C)CC
2-methylbutane	CC(C)CC
Isobutane	CC(C)C
Decane	CCCCCCCCCC
Dodecane	CCCCCCCCCCCC
Hexadecane	CCCCCCCCCCCCCCCC
2-methylpentane	CC(C)CCC
4-methyl-3-ethyl-heptane	CCC(CC)C(C)CCC
5-methylnonane	CCCCC(C)CCCC
5-ethylnonane	CCCCC(CC)CCCC
2-methyl-4-ethyl-nonane	CC(C)CC(CC)CCCC
23-dimethyl-3-isopropylhexane	CC(C)C(C)(C(C)C)CCC
23-dimethyl-3-ethylhexane	CC(C)C(CC)(C)CCC
249-trimethyl-7-isopropyldecane	CC(C)C(C)CCCC(C(C)C)CC(C)C
226-trimethyl-3-ethyloctane	CC(C)(C)C(CC)CCC(C)CC
28-dimethyl-6-propyl-decane	CC(C)CCCC(CCC)CC(C)CC
235-trimethyl-3-ethyl-4-propylheptane	CC(C)C(C)(CC)C(CCC)C(C)CC
36-dimethyl-3-ethyloctane	CCC(CC)(C)CCC(C)CC
224-trimethyl-5-isopropyloctane	CC(C)(C)CC(C)C(C(C)C)CCC
24-dimethylpentane	CC(C)CC(C)C
24-dimethyl-3-ethylhexane	CC(C)C(CC)C(C)CC

Table S2: Alkanes considered (continued)

Name	SMILES
45-dimethyl-3-ethylheptane	CCC(CC)C(C)C(C)CC
27-dimethyl-3-isopropyloctane	CC(C)C(C(C)C)CCCC(C)C
238-trimethyldecane	CC(C)C(C)CCCCC(C)CC
244-methyl-3-isopropylhexane	CC(C)C(C(C)C)C(C)(C)CC
26-dimethyl-4-isopropylheptane	CC(C)CC(C(C)C)CC(C)C
23-dimethyl-34-diethylpentane	CC(C)C(CC)(C)C(CC)CC
45-dimethyl-35-diethyloctane	CCC(CC)C(C)C(CC)(C)CCC
247-trimethyl-6-ethylnonane	CC(C)CC(C)CC(CC)C(C)CC
2356-tetramethylheptane	CC(C)C(C)CC(C)C(C)C
3445-tetramethyl-4-isopropylnonane	CCC(C)C(C(C)C)C(C)(C)C(C)CCC
334-trimethyl-77-dimethylnonane	CCC(C)(C)C(C)CCC(CC)(CC)CC
5-propylnonane	CCCCC(CCC)CCCC
4-isopropylnonane	CCCC(C(C)C)CCCCC
7-isopropyltridecane	CCCCCCC(C(C)C)CCCCCC
6-propyltridecane	CCCCCC(CCC)CCCCCCC
22-Dimethylhexane	CC(C)(C)CCCC

Model performance - hyperparameters Bayesian optimisation results

The SVR, GBC/GBR and XGB models are influenced by a set of training parameters,[2–4] such as kernel width (for the SVR) or the tree depth (for GBR and XGB). The following tables show what hyperparameters are optimized prior to training, as well as the boundaries and optimal values.

SVR hyperparameters

The optimal kernel for the Support Vector Regressor was the radial-basis function kernel. This results in 3 hyperparameters that need to be tuned:[2,3] the regularization parameter, the error margin (from the SVR model itself), and the kernel width. These are subjected to upper and lower bounds for the optimization, and have an optimal value for the training data. All of these can be consulted in table S3.

Table S3: SVR hyperparameters. Upper and lower bounds of optimization are supplied, as well as their optimal values for minimizing the mean-square error during fitting.

Hyperparameter	Upper bound	Lower bound	Optimal
Regularization parameter	100	0.1	72.1593
Kernel width	10^{-4}	1	0.0415
Error margin	10^{-3}	1	0.0196

GBR hyperparameters

To tune the Gradient-Boosted trees, eight hyperparameters are optimized for:[2,5] the number of trees, the maximum depth of each tree, the learning rate (constant with which the learning from pseudo-residuals is amplified), the subsample fraction (number between 0 and 1 quantifying how much of the training data is used to train the tree), the necessary amount of values for a split, the target amount of values in each leaf, the minimum weighted fraction of a leaf and the maximum amount of features used (bounded between 0 and 1, as a fraction). These are subjected to upper and lower bounds for the optimization, and have an optimal value for the training data. The hyperparameter values for the classifier stage of the cascade can be seen in table S4and the hyperparameter values for the model used as a. regressor can be found in table S5

Table S4: GBC hyperparameters for the classifier. Upper and lower bounds of optimization are supplied, as well as their optimal values for minimizing the mean-square error during fitting.

Hyperparameter	**Upper bound**	**Lower bound**	**Optimal value**
Number of trees	400	50	238.3521
Maximum depth of each tree	15	1	12.7519
Learning rate	2	0.001	0.0711
Maximum number of leaves	50	2	32.0234
Subsample fraction	1.0	0.4	1.0000
Minimum of values for a split	20	2	17.4221
Minimum of values in a leaf	10	1	5.2182
Minimum weighted fraction leaf	0.4	0.0	0.0000
Maximum amount of features used	1.0	0.3	0.7835

Table S5: GBT hyperparameters for the regressor. Upper and lower bounds of optimization are supplied, as well as their optimal values for minimizing the mean-square error during fitting.

Hyperparameter	**Upper bound**	**Lower bound**	**Optimal value**
Number of trees	400	50	298.8817
Maximum depth of each tree	15	1	2.1969
Learning rate	2	0.001	0.5801
Maximum number of leaves	50	2	34.4959
Subsample fraction	1.0	0.4	0.7618
Minimum of values for a split	20	2	3.8681
Minimum of values in a leaf	10	1	9.9248
Minimum weighted fraction leaf	0.4	0.0	0.0496
Maximum amount of features used	1.0	0.3	0.4045

XGB hyperparameters

The XGBoost extreme gradient-boosted tree uses some similar parameters as the gradient-boosted tree, but leaf and leaf weight-related parameters are concatenated in a minimum loss reduction, a minimum child weight, as well as L1 and L2 regularization parameters. This is because the complexity is penalized when creating the boosting trees.[4] These are subjected to upper and lower bounds for the optimization, and have an optimal value for the training data. All of these are listed in table S6.

Table S6: XGBoost hyperparameters. Upper and lower bounds of optimization are supplied, as well as their optimal values for minimizing the mean-square error during fitting.

Hyperparameter	**Upper bound**	**Lower bound**	**Final value**
Number of trees	800.0000	50.0000	217.0141
Maximum depth of each tree	12.0000	2.0000	12.0000
Learning rate	0.3000	0.0001	0.3000
Subsample fraction	1.0000	0.1000	1.0000
Column subsample	1.0000	0.5000	0.5000
Minimum loss reduction	5.0000	0.0000	0.0000
Minimum child weight	20.0000	1.0000	1.1569
L1 regularization	10.0000	0.0000	10^{-8}
L2 regularization	10.0000	0.0000	6.8472

Robustness study - random seed test raw results

The table below contains the resulting values for the $\mathrm{MACRO} - f1$ (i.e. arithmetic average of the class-wise $f1$ scores, see Eq. 7 of the main text) of the classifier and R^2 of the regressor for a certain random seed given to the cascade. As it can be seen, the values remain mostly consistent throughout, even if the regressor has a few outliers. This is further confirmed by the low standard deviation.

Table S7: Robustness study - raw results of random seed test. 30 random seeds are used for the model, and the resulting classifier MACRO $-$ $f1$ and regressor R^2 are recorded. The results show that not only is the model reliable against randomness in data splits and initializations (since values change very little), but also that the expected performance is expected to be quite high no matter what.

Random seed/[-]	MACRO $-$ **f1**/[-]	R^2/[-]
860	0.993	0.979
200	0.985	0.966
412	0.993	0.969
210	0.986	0.954
767	0.994	0.954
578	0.991	0.938
750	0.995	0.979
970	0.994	0.943
370	0.989	0.910
497	0.991	0.957
597	0.991	0.933
71	0.994	0.956
270	0.994	0.933
557	0.988	0.953
528	0.994	0.972
911	1.000	0.966
824	0.992	0.948
563	0.987	0.968
296	0.985	0.968
424	0.983	0.945
302	0.991	0.960
625	0.995	0.967
896	0.993	0.965
188	0.985	0.948
24	0.989	0.965
858	0.990	0.918
447	0.988	0.957
745	0.992	0.961
575	0.994	0.954
216	0.988	0.972
Mean ± standard deviation	0.991 ± 0.00385	0.955 ± 0.0165

Robustness study - raw results test size fraction test

The tables below contains the resulting values for the $\mathrm{MACRO} - \mathbf{f1}$ of the classifier and $\mathbf{R^2}$ of the regressor for a certain random seed given to the cascade. To make it more manageable, each table represents a test size fraction.

Table S8: Robustness study - raw results of random seed test (Test size fraction = 0.1)

Random seed/[-]	$\mathrm{MACRO} - \mathbf{f1}$/[-]	$\mathbf{R^2}$/[-]
248	0.991	0.949
336	0.995	0.936
785	0.987	0.969
735	1.000	0.926
504	0.997	0.956
805	1.000	0.988
578	0.997	0.934
20	0.994	0.959
837	0.994	0.962
480	0.984	0.985
161	1.000	0.968
709	0.994	0.945
87	0.997	0.945
590	0.993	0.974
436	0.991	0.915

Table S9: Robustness study - raw results of random seed test (Test size fraction = 0.2)

Random seed/[-]	$\mathrm{MACRO} - \mathbf{f1}$/[-]	$\mathbf{R^2}$/[-]
248	0.991	0.941
336	0.991	0.956
785	0.986	0.968
735	0.991	0.956
504	0.997	0.956
805	0.997	0.971
578	0.991	0.938
20	0.995	0.967
837	0.986	0.947
480	0.991	0.955
161	0.985	0.949
709	0.992	0.949
87	0.992	0.968
590	0.994	0.969
436	0.989	0.952

Table S10: Robustness study - raw results of random seed test (Test size fraction = 0.3)

Random seed/[-]	$\mathrm{MACRO} - \mathbf{f1}$/[-]	$\mathbf{R^2}$/[-]
248	0.991	0.946
336	0.986	0.954
785	0.990	0.962
735	0.987	0.948
504	0.995	0.956
805	0.994	0.959
578	0.986	0.946
20	0.992	0.957
837	0.990	0.953
480	0.986	0.955
161	0.985	0.943
709	0.984	0.958
87	0.985	0.956
590	0.987	0.955
436	0.988	0.955

Table S11: Robustness study - raw results of random seed test (Test size fraction = 0.4)

Random seed/[-]	MACRO − **f1**/[-]	$\mathbf{R^2}$/[-]
248	0.991	0.954
336	0.983	0.955
785	0.987	0.965
735	0.984	0.953
504	0.994	0.946
805	0.986	0.960
578	0.984	0.941
20	0.988	0.958
837	0.989	0.946
480	0.985	0.955
161	0.991	0.927
709	0.985	0.953
87	0.987	0.952
590	0.991	0.954
436	0.988	0.955

Table S12: Robustness study - raw results of random seed test (Test size fraction = 0.5)

Random seed/[-]	MACRO − **f1**/[-]	$\mathbf{R^2}$/[-]
248	0.981	0.984
336	0.985	0.947
785	0.981	0.952
735	0.981	0.957
504	0.988	0.941
805	0.988	0.953
578	0.985	0.941
20	0.984	0.953
837	0.987	0.944
480	0.984	0.930
161	0.976	0.941
709	0.983	0.955
87	0.980	0.948
590	0.986	0.950
436	0.988	0.948

References

(1) Baerlocher, C.; McCusker, L. B.; Brouwer, D.; Marler, B. Database of Zeolite Structures. `https://www.iza-structure.org/databases/`, 2025; Accessed: 2025-09-01.

(2) Guido, S.; Mueller, A. C. *Introduction to machine learning with python*; O'Reilly Media, 2016; p 392.

(3) Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer New York, 2009.

(4) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **2016**, 785–794, Also available as arXiv:1603.02754.

(5) Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* **2001**, *29*, 1189–1232.