Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Enhanced Thermophysical Property Prediction with Uncertainty Quantification using Group Contribution-Gaussian Process Regression†

Barnabas P. Agbodekhe,[a] Montana N. Carlozo,[a] Dinis O. Abranches,[b] Kyla D. Jones,[a] Alexander W. Dowling,*[a] and Edward J. Maginn*[a]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

Group contribution (GC) models are powerful, simple, and popular methods for property prediction. However, the most accessible and computationally efficient GC methods, like the Joback and Reid (JR) GC models, often exhibit severe systematic bias. Furthermore, most GC methods do not have uncertainty estimates associated with their predictions. The present work develops a hybrid method for property prediction that integrates GC models with Gaussian process (GP) regression. Predictions from the JR GC method, along with the molecular weight, are used as input features to a GP model, which learns and corrects the systematic bias in the GC predictions, resulting in highly accurate property predictions with reliable uncertainty estimates. The method was applied to six properties; normal boiling temperature (T_b), enthalpy of vaporization at T_b (ΔH_{vap}), normal melting temperature (T_m), critical pressure (P_c), critical molar volume (V_c), and critical temperature (T_c). The CRC Handbook of Chemistry and Physics was used as the source of experimental data. The final collected experimental data ranged from 485 molecules for ΔH_{vap} to 5640 for T_m. The proposed GCGP method was found to provide much improved property prediction accuracy compared to the GC-only method. The coefficient of determination (R^2) values of the testing set predictions are $\geq$ 0.85 for five out of six and $\geq$ 0.90 for four out of six properties modeled, and compare favorably with other methods in the literature. The uncertainty estimates were reliable, and the GCGP method proved robust to variations in GP model architecture and kernel choice.

1 Introduction

The discovery of new materials is a cornerstone of sustainability research, particularly in addressing global challenges such as climate change[1,2], energy efficiency[3], environmental preservation[4], and health[5]. A timely and important example of this falls within the field of cooling and refrigeration[6–8]. The search for environmentally friendly alternative refrigerants[9]and materials for refrigerant recycling[10–14] has become a critical area of research. Other areas of active research that require the discovery of molecules include small-molecule drug discovery[5], the design of environmentally benign solvents[15], and the development of materials for energy sustainability[3].

The discovery and development of new materials to meet these challenges requires the reliable prediction of material properties. Experimental exploration of all possible molecules and properties needed for any material discovery problem is often not feasible. Databases[16–19] of materials and some of their experimentally measured properties have been assembled for decades. However, these databases contain a minuscule fraction of potentially relevant molecules. Furthermore, assuming that large enough databases of potential molecules are available or developed for the discovery of materials, the properties required to assess the suitability of materials are not always available[20]. Predictive computational tools are essential for streamlining the process of molecule discovery. Computer-aided molecular design (CAMD) is a well-established molecular discovery method that integrates and automates considerations from molecular to process scales in the development of new materials and processes[21]. It has key advantages over traditional database screening methods, such as the potential to discover new molecules not present in compiled databases. However, one of the persistent challenges is the availability and integration of fast and reliable property prediction

[a] Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: ed@nd.edu, adowling@nd.edu
[b] CICECO - Aveiro Institute of Materials, Department of Chemistry, University of Aveiro, 3810-193 Aveiro, Portugal.

† Supplementary Information available: [The Supplementary Information is available free of charge and includes further information on data collection and preparation, data analysis, final model parameters, and additional results on a variety of tests conducted in this work. All codes and final results are also available on the project's GitHub repository]. See DOI: 00.0000/00000000.

* Corresponding authors

methods in CAMD workflows[21].

Group contribution (GC) models have long been used to predict the properties of materials within CAMD and other material discovery workflows, particularly to estimate thermophysical properties[22–26]. GC models operate by decomposing molecular structures into predefined functional groups and assigning specific contributions or interaction parameters to each group on the basis of experimental data.

Substantial effort has been made to develop GC-based thermodynamic models, including equation of state (EoS) and activity coefficient (AC) models. Examples of GC-based EoS models include the Predictive Soave-Redlich-Kwong (PSRK)[27,28], GC-SAFT[29–31], and SAFT-γ-Mie[32–35] models, amongst others. An example of a GC-based AC model is the UNIQUAC[36] Functional-group Activity Coefficients (UNIFAC) model[37]. These GC-based EoS or AC models are of great utility in CAMD, particularly for predicting thermodynamic properties of mixtures across a wide range of temperatures, pressures, and compositions[38–41]. However, implementing these models can be cumbersome, and their computational efficiency is often limited due to the need to evaluate complex derivatives[42].

An alternative class of GC methods is the class of semi-empirical or correlation-based GC models. These GC methods typically consist of several models or equations—one equation for one property—for direct and efficient computation of properties without the need to evaluate complex derivatives of other properties, as is required in EoS models. Notable examples of such semi-empirical GC models include the Joback and Reid (JR) method[43], the Lydersen method[44], and the Marrero-Gani method[45] amongst others[46]. These models are particularly useful for material screening tasks that involve pure fluids. Therefore, these methods can be applied, at least in a preliminary stage, to many material screening and CAMD tasks[21]. Their simplicity and generalizability make them invaluable tools for screening chemical systems and designing processes without requiring extensive experimental datasets. Because property predictions using these types of GC models do not rely on calculating the derivatives of other thermodynamic properties, they offer the advantages of speed and ease of implementation compared to other methods. Compared to GC-based thermodynamic models, these types of GC models are also more generalizable for predicting diverse properties, such as environmental[47,48] or safety properties[49–53] of materials.

However, as highlighted in recent studies[54], limitations in available group parameters and interaction data often restrict the predictive accuracy and scope of GC models. Furthermore, the most accessible types of these GC methods, which are first-order GC models such as the JR GC method[43], are known to have significant systematic bias[55,56]. Moreover, common GC models generally do not have uncertainty estimates associated with their predictions, which is essential for material screening.

The emergence of machine learning (ML) techniques has opened new avenues for addressing some of the limitations of GC approaches. ML methods can predict the properties of molecules with high accuracy by leveraging large datasets and advanced algorithms such as neural networks (NNs)[57], support vector (SV) machines[58,59], Gaussian process (GP) models[60–62], random forests[63], boosting algorithms[64–67], and so on, enabling rapid virtual screening of chemical candidates.

However, ML models have several drawbacks. They typically require a large amount of data, which is not always available[68,69]. Also, unlike traditional thermodynamic models and some GC models, ML models rarely have clear physical interpretability[70,71]. Furthermore, uncertainty propagation and estimation from complex ML techniques, such as deep neural networks, can be cumbersome[72]. GP ML surrogate models, in contrast, are well-suited for applications with limited data and inherently include uncertainty quantification. The drawbacks of GPs include scalability to large datasets with many observations or many input features, difficulty scaling to multiple outputs, and challenges approximating discontinuous functions[73].

Despite these limitations, ML offers powerful tools for identifying patterns and correlations in complex, multidimensional datasets, which can be leveraged to extend the applicability and accuracy of GC models. For instance, matrix completion methods have been used to predict missing group interaction parameters in thermodynamic GC models[54], demonstrating how data-driven approaches can fill gaps in traditional GC model parameterization.

A few works have explored the synergistic benefits of combining GC and ML for enhanced property predictions. One of these efforts includes using the number of functional groups along with several properties such as T_c and P_c as inputs to several ML models for predicting triple point temperature[74]. Some other works[75–79] have explored the combination of GC and ML for property prediction. Xinyu et al.[79] used inputs from a third-order GC-based fragmentation as features to train SV and GP models. This resulted in models with a 424-dimensional input size. These previous attempts at combining GC and ML for enhanced property predictions have primarily focused on utilizing GC-based molecular fragmentations as molecular descriptors in ML models for property prediction[74,75,79]. This leads to high-dimensional, discrete input feature spaces for training ML models. Such high-dimensional input spaces pose severe challenges for certain ML algorithms, such as GP regression (GPR)[73,80]; thus, most of the GC-ML methods in the literature have focused on ML methods such as SV regression, boosting algorithms, and NNs, which do not provide a convenient route for reliable prediction uncertainty quantification.

The present work aims to enable the efficient, reliable, and parsimonious prediction of thermophysical properties with uncertainty quantification by combining the strengths of simple, first-order, semi-empirical GC methodologies and GP models. We use property predictions from a simple first-order GC method (the JR GC model), along with a readily accessible molecular property (molecular weight), as the only two inputs to the GP. The GC predictions often have significant systematic biases for several properties, which are then corrected by training the GP.

By integrating the systematic framework of GC models with the predictive power of GPR, we propose a hybrid approach that overcomes existing data limitations, improves predictive accuracy compared to GC-only methods, provides uncertainty estimates, requires two simple-to-compute input features, provides inter-

pretability, and maintains computational efficiency. Our study evaluates the performance of this hybrid model approach[81,82] in comparison to predictions made using only the GC model. The approach aims to provide a versatile and robust framework for property prediction, enabling the design and optimization of a broad range of chemical systems.

2 Methods and Data

This work demonstrates the proposed method, considering up to 5640 organic molecules that encompass various classes of organic compounds. The methods and data collection are described below.

2.1 Data Collection and Preparation

Six properties were modeled in this work: normal boiling temperature (T_b), enthalpy of vaporization at T_b (ΔH_{vap}), critical pressure (P_c), critical molar volume (V_c), critical temperature (T_c), and the normal melting temperature (T_m). These properties are essential for several materials discovery tasks. T_b, for example, is used in several engineering correlations and models to predict properties such as the enthalpy of vaporization at temperatures other than the normal boiling temperature[83]. In the JR GC method, T_b is used to compute T_c[43]. T_b is also commonly used to calculate the acentric factor of molecules, which is correlated with other properties such as the liquid heat capacity[83,84]. T_c, P_c, and V_c are essential for the consideration of stability, safety, and the determination of appropriate operating regions for new fluids[85]. They are also used to estimate parameters for equations of state. ΔH_{vap} is generally important for any material design task for applications that involve a phase change between the liquid and vapor phases, such as refrigeration[83,86]. T_m is important for applications in which the solid-liquid phase transition is an important consideration[87]. Furthermore, these properties were selected as non-temperature-dependent properties to demonstrate the GCGP method.

Three types of data are collected or computed for each molecule and property to build the complete datasets used in this work: experimental property data, the JR GC property predictions, and the molecular weights (MW).

2.1.1 Experimental Data Collection

Unless otherwise noted, the experimental data for training GP models were obtained from the 105th edition of the CRC Handbook of Chemistry and Physics[18]. As described later, some experimental data for ΔH_{vap} were collected from Yaws' Critical Property Data for Chemical Engineers and Chemists[17]. For each property, data points were collected for all molecules that could be treated with the JR GC method and for which there were experimental data from the CRC Handbook of Chemistry and Physics.

Table 1 shows the total number of experimental data points used in this work for each property. T_m had the highest number of data points for molecules whose melting temperature could be predicted using the JR GC model, while ΔH_{vap} had the least. 514 and 416 experimental data points for T_m and T_b, respectively, in the CRC Handbook of Chemistry and Physics were omitted from this study (and thus not included in Table 1), as flags were provided within the database to indicate that the reported temperatures may not be the true melting or boiling temperatures, as the molecules could undergo decomposition or sublimation at those temperatures.

Table 1 Amount of Final Collected Data

Property	Total data points	Training set	Testing set
T_b	4321	3457	864
ΔH_{vap}	485	388	97
P_c	684	547	137
V_c	698	558	140
T_c	712	570	142
T_m	5640	4512	1128

2.1.2 Joback and Reid GC Predictions

The JR GC method is a first-order GC method presented in equations 1 - 6 for the six properties considered. The model parameters are available in the original work[43]. The JR GC method was selected for this work due to its popularity, ease of use, accessibility, and availability of open source software (e.g., JRgui[88]), for automatic generation of JR GC predictions of molecules given their SMILES strings.

$$T_b\,[\mathrm{K}] = 198.2 + \sum_{i\in\mathscr{G}} n_i \times T_{b,i} \tag{1}$$

$$H_{vap}\,[\mathrm{kJ/mol}] = 15.30 + \sum_{i\in\mathscr{G}} n_i \times H_{vap,i} \tag{2}$$

$$P_c\,[\mathrm{bar}] = \left[0.113 + 0.0032N_a - \sum_{i\in\mathscr{G}} n_i \times P_{c,i}\right]^{-2} \tag{3}$$

$$V_c\,[\mathrm{cm^3/mol}] = 17.5 + \sum_{i\in\mathscr{G}} n_i \times V_{c,i} \tag{4}$$

$$T_c\,[\mathrm{K}] = T_b\left[0.584 + 0.965\sum_{i\in\mathscr{G}} n_i \times T_{c,i} - \left(\sum_{i\in\mathscr{G}} n_i \times T_{c,i}\right)^2\right]^{-1} \tag{5}$$

$$T_m\,[\mathrm{K}] = 122.5 + \sum_{i\in\mathscr{G}} n_i \times T_{m,i} \tag{6}$$

In the above equations, n_i is the number of structural units of type i in the molecule. $\mathscr{G}$ is the set of groups with parameters in the JR GC model. $T_{b,i}, H_{vap,i}, ..., T_{m,i}$ are the JR GC parameters for the structural unit (group) i for each property. These parameters determine how the presence of each structural unit changes or contributes to the properties.

The JR GC method works by dividing the molecule into predefined structural units, for which parameters are available in the JR GC method. The desired property of the molecule is then predicted using the appropriate JR GC equation from equations 1 - 6. The parameters for these equations are tabulated. Supplementary Information (SI) Figure S1 shows an example of how the JR GC method is used to compute properties.

In this work, the JRgui[88] software, an open-source Python-based code, was used to automatically compute the JR GC pre-

dictions for all properties using the SMILES strings of molecules. SMILES strings that cannot be treated using the JR GC method were filtered out. The SMILES strings were obtained by parsing the Chemical Abstracts Service (CAS) registry numbers of the molecules in the CRC Handbook of Chemistry and Physics using PubChemPy (version: 1.0.4)[89], another open source Python-based package for interfacing with the PubChem[16] database of compounds. The PubChem database contains over 100 million compounds and contains SMILES strings for all or almost all compounds for which it has an entry. In this work, we assume that all or almost all of the molecules in the CRC Handbook of Chemistry and Physics will have an entry in the PubChem database, and thus have their SMILES strings available from PubChem.

The JRgui software also provides the values of more than 200 molecular descriptors from RDKit (version: 2020.09.1.0)[90] in addition to other output data. The molecular weight (MW) is one of the outputs of the JRgui tool and was used as the source of MW data for this work. Note that MW can be readily computed in the same fashion as some other properties from simple GC equations by simply summing the molecular weights of the structural units in a molecule, so there is no need to use RDKit, JRgui, or any specialized tool.

We found that the JRgui software failed to correctly parse SMILES of co-crystals, molecular complexes, ring-embedded tertiary amines, and anhydrides, which are not amenable to the JR GC method. The JRgui software outputted incorrect JR GC properties due to erroneous molecular fragmentation for these classes of molecules. We applied additional filtering to remove data corresponding to such molecules to obtain the final data sets used in this work.

2.2 Data Pre-processing

We present below some details of the data pre-processing steps, including data quality checks, data analysis, and visualization to aid model building.

2.2.1 Data Quality

We performed a basic two-dimensional outlier detection analysis using the JR GC predictions and collected experimental data. (See SI Section S1.2 for details.) We observed certain data points that showed significant deviations from the general trends in the JR GC predictions compared to experiments for ΔH_{vap} and flagged these points as 'outliers' with respect to the JR GC model (see SI Figures S2 and S3). In further investigation of these points, we identified three experimental ΔH_{vap} values for which the CRC Handbook of Chemistry and Physics had incorrect data entries. These molecules are butyrolactone, 1-methylcyclohexanol, and (+)-2-Bornanone with CAS registry numbers 96-48-0, 590-67-0, and 464-49-3, respectively (see SI Figure S4). We ascertained that the data entries for these molecules were incorrect by comparing them against data from the Yaws' Critical Property Data for Chemical Engineers and Chemists[17] as available in the Knovel database and data from the National Institute of Standards and Technology (NIST)[91] Webbook. The NIST Webbook and Yaws' Critical Property Data agreed with each other, while the CRC Handbook data differed for these three molecules. Furthermore, once the experimental ΔH_{vap} data for these three molecules were replaced with those from the Yaws' Critical Property Data, they ceased to be flagged by our outlier detection procedure (See SI Figure S5). The other data points that were flagged as 'outliers' for ΔH_{vap} were found to be due to limitations in the parameterization of the JR GC method (see SI Figures S6a-c). This is discussed in more detail in a later section.

We note that the T_m data collected from the CRC Handbook of Chemistry and Physics had several entries for which the T_m values were exactly the same, even for molecules with widely differing structural units, functional groups, and molecular weight. We performed spot checks and compared some of the T_m data collected from the CRC Handbook of Chemistry and Physics with those from the Yaws' Critical Property Data for Chemical Engineers and Chemists[17] as available in the Knovel database. We found that the entries in the CRC handbook of Chemistry and Physics agree with those from the Yaws' Critical Property Data for Chemical Engineers and Chemists. T_m poses an interesting challenge, considering that molecules with seemingly very different functional groups and molecular structures have similar values of T_m. See Section 3.1 for further discussion.

2.2.2 Data Analysis and Demonstration of Systematic Bias in JR GC Predictions

We begin by analyzing the trends in the data. Figures 1 and 2 demonstrate that the JR GC predictions and molecular weight are related to the experimental data for all properties of interest. Figure 1 shows that the JR GC predictions and the experimental data are fairly linearly correlated for ΔH_{vap}, P_c, and V_c. The JR GC models for T_m, T_b , and T_c are much worse predictors of the experimental data as quantified in Sections 3.1 and 3.2. Thus, we observe a non-linear trend indicated by non-zero trends in the discrepancy, as shown in Figure 2.

Figure 2 shows a relationship between molecular weight and the experimental data and JR GC predictions for V_c, T_b, T_c, and T_m. We observe that the discrepancy in ΔH_{vap} does not have a strong correlation with molecular weight (i.e. there is no clear discrepancy color gradient with changing MW), and that P_c exhibits a strong nonlinear trend suggesting that molecular weight is an excellent molecular descriptor for P_c and a subpar descriptor for ΔH_{vap} (see SI Figures S7 and S8 in SI section S1.2.2).

The systematic bias in T_m, T_b, and T_c highlights shortcomings of the JR GC method, which assumes that structural units contribute to the value of these properties monotonously. We observe, for example, that the JR GC method predicts that several organic molecules would have values of T_m greater than 1500 K, which is not the case in nature. Molecules — even within the same family — do not monotonously and boundlessly melt at higher temperatures as they get bigger. The systematic bias of the JR GC predictions for ΔH_{vap} and P_c is more nuanced. Unlike the other properties for which the JR GC method shows a systematic bias that is generally correlated with molecular weight, the systematic bias of the JR GC method for ΔH_{vap} and P_c is for specific classes of molecules.

In Figure 1 there are two points (a and b) with conspicuously low JR GC ΔH_{vap} predictions, which correspond to highly fluori-

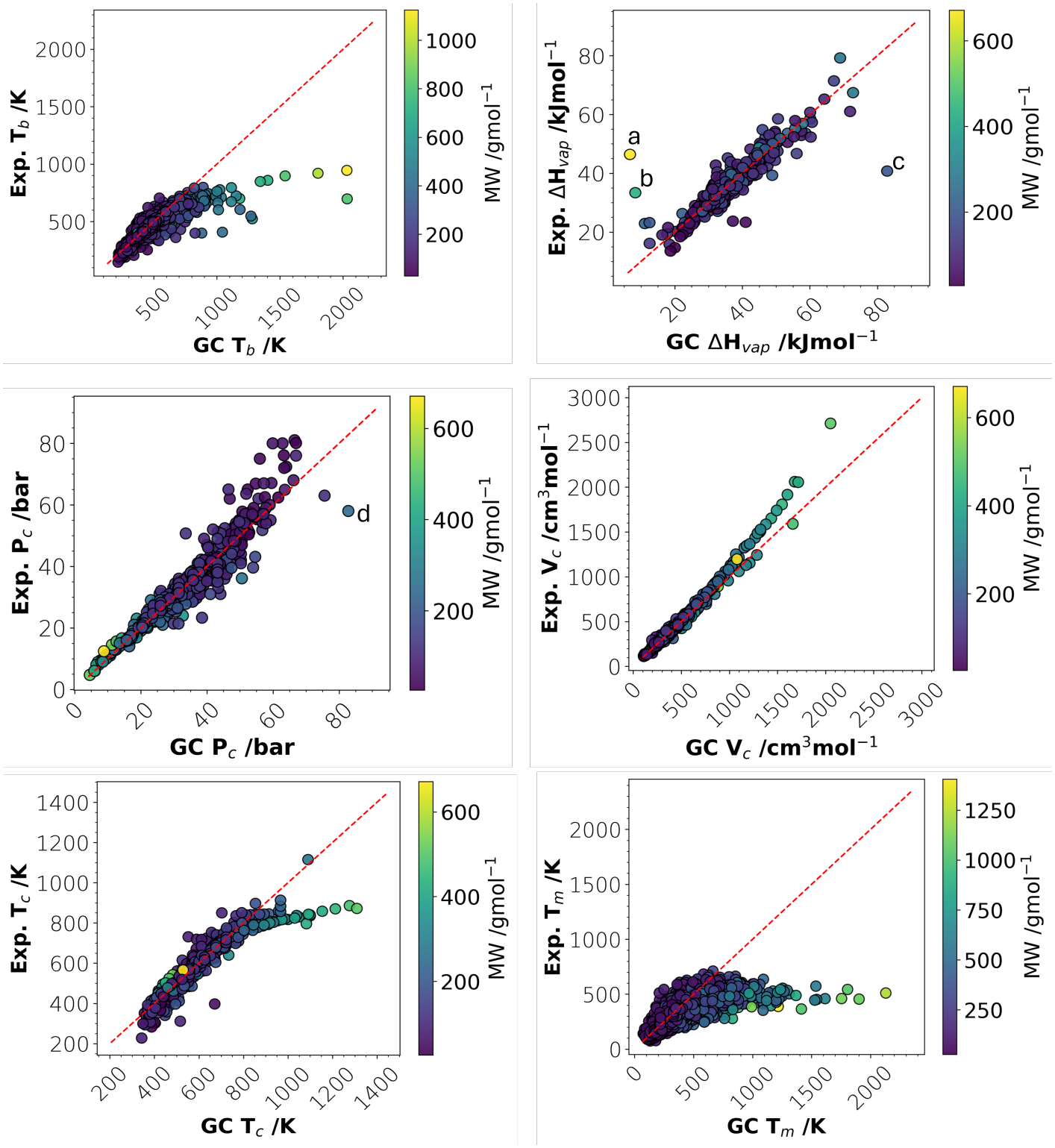

Fig. 1 2D visualization of JR GC predictions against experimental values. Points a, b, c, and d correspond to molecules for which the JR GC method shows large deviations compared to experimental data for ΔH_{vap} (a, b, and c) and P_c (d). Points a and b correspond to highly fluorinated molecules, points c and d correspond to a highly nitrated molecule and a highly brominated molecule, respectively.

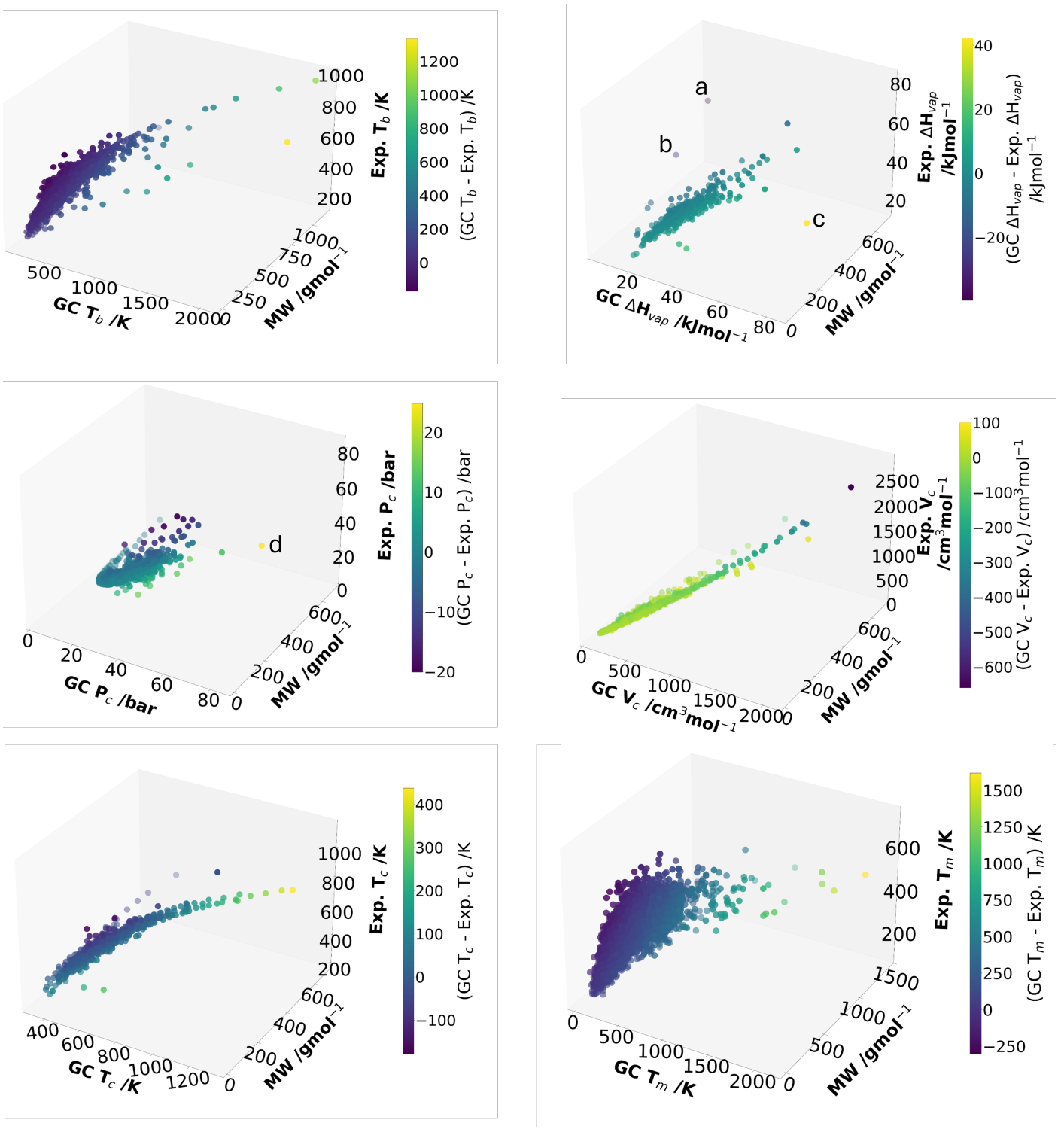

Fig. 2 3D visualization of JR GC predictions against experimental values and MW. Points a, b, c, and d are as previously discussed.

nated molecules with moderate to high MW. The two molecules with this large underestimation in ΔH_{vap} using the JR GC method are shown in SI Figures S6a and S6b. The contribution of the fluorine group to ΔH_{vap} according to the JR GC method is -0.67 kJ/mol. This represents the only negative value in the parameter set for ΔH_{vap} in the JR GC method; all other groups have positive contributions to ΔH_{vap} in the JR GC method[43]. This explains why, for highly fluorinated molecules, the JR GC method predicts very low values of ΔH_{vap} contrary to experimental values. The JR GC method could predict negative ΔH_{vap} values for sufficiently fluorinated molecules, which would be unphysical.

SI Figure S6c shows another class of molecules for which the JR GC method has a large systematic bias in its ΔH_{vap} predictions. They are highly nitrated compounds, such as tetranitromethane shown in SI Figure S6c. The JR GC ΔH_{vap} prediction for tetranitromethane is 82.89 kJ/mol and can be observed in Figure 1 as the highest JR GC ΔH_{vap} prediction (point c) in our data. The JR GC method predicts that every -NO_2 structural unit in a molecule should contribute 16.738 kJ/mol to the ΔH_{vap} of the molecule. This contribution is much higher than those of most other structural units in the JR GC method parameter set for ΔH_{vap}. This leads to an overestimation in ΔH_{vap} for highly nitrated molecules. A similar scenario is observed for JR GC P_c predictions for highly brominated molecules (point d in Figure 1). The molecules corresponding to points a-d were included in the training set for model development using stratified sampling discussed in Section 2.4. In summary, Figure 2 visualizes the 3D relationship between the input features, MW, and JR GC prediction, with experimental data for all properties.

2.3 GP Modeling

We tested several aspects of the implementation details of the GP model and discussed how these details impact the results. It is therefore important to provide some background information on the methods used.

We start by establishing notation. We define the dataset $\mathscr{D}_p := \{(y_{\mathrm{GC}_i}, \mathrm{MW}_i), y_{\mathrm{exp}_i})\}_{i=1}^n$ for all molecules n and each property $p \in \mathscr{P} := \{H_{\mathrm{vap}}, P_c, T_c, T_b, T_M, V_c\}$ of interest. We define the vector $\mathbf{y}_{\mathrm{exp}} = [y_{\mathrm{GC}_i} | \forall\ i \in \{1, ..., n\}]$ for each property, where p is dropped for convenience. Similarly, we define the input feature vector $\mathbf{x}_i = [y_{\mathrm{GC}_i}, \mathrm{MW}_i]$ which are stacked vertically to form the input feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where $d = 2$. Our goal is to train GP models to predict $\mathbf{y_{exp}}$ based on the inputs $\mathbf{X}$.

2.3.1 Gaussian Process Basics

A stochastic process is a (infinite) collection of random variables indexed by a set $\{\mathbf{x}\}$. A GP is a stochastic process in which any finite number of random variables have a joint Gaussian distribution[73]. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote an index and $f : \mathbb{R}^d \to \mathbb{R}$ denote a random variable that is indexed by $\mathbf{x}$ (i.e., the stochastic process). A GP is specified by a mean function

$$m(\mathbf{x}) := \mathbb{E}[f(\mathbf{x})] \tag{7}$$

and a covariance function

$$k(\mathbf{x}, \mathbf{x}') := \mathrm{Cov}[f(\mathbf{x}), f(\mathbf{x}')]. \tag{8}$$

The notation $f(\mathbf{x}) \sim \mathscr{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ denotes that $f(\cdot)$ follows a GP distribution with mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$. Equivalently, by the definition of a GP, for any finite subset $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of the random variables, $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^\intercal$ follows a multivariate normal distribution with a mean vector and covariance matrix governed elementwise by equations 7 and 8, respectively. That is, $\mathbf{f} \sim \mathscr{N}(\boldsymbol{\mu}, \mathbf{K})$, where $\boldsymbol{\mu} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n))^\intercal$ and

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}. \tag{9}$$

In Bayesian nonparametric statistics, a GP is used as a prior for a random variable indexed by an infinite set. Upon observing a finite subset of these random variables, the posterior is another GP. This is commonly applied in regression settings to recover latent functions. See relevant texts[73,92] for a more complete introduction to GPs.

2.3.2 Model Selection and Kernels

When deploying GPs for regression, (lack of) prior information of the latent function is encoded through the mean and covariance functions. The mean function represents prior belief about the average value of the function being modeled. It sets the baseline for the GP before any data are observed. This section focuses on how to choose stationary kernel functions for modeling the covariance of the GP that are common in application literature. See Genton[93] for a more generalized perspective on classes of kernel functions.

A *kernel* refers to a function that defines a similarity measure between pairs of points. In the context of GPs, a kernel is a positive-definite function that defines the covariance structure. For example, the squared exponential (SE) (i.e., Gaussian) kernel is given by

$$k_{\mathrm{SE}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\mathbf{r}^\intercal \boldsymbol{\Lambda}^{-1} \mathbf{r}\right), \tag{10}$$

where $\mathbf{r} = \mathbf{x}_i - \mathbf{x}_j$ is the distance between two points, σ_f^2 is the variance of the process, and $\boldsymbol{\Lambda}$ is a matrix of length scales that control the smoothness of the function. The SE kernel assumes the underlying function is infinitely differentiable. Thus, the SE kernel is widely used due to its ability to model smooth functions. Furthermore, a modeler can structure the length scale matrix $\boldsymbol{\Lambda}$ to encode additional smoothness assumptions of the underlying function[73]. This is covered in detail at the end of this section.

A more general form of equation 10 is the rational quadratic (RQ) kernel given by

$$k_{\mathrm{RQ}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left(1 + \frac{1}{2\alpha}\mathbf{r}^\intercal \boldsymbol{\Lambda}^{-1} \mathbf{r}\right)^{-\alpha}. \tag{11}$$

The RQ kernel can model a wider range of functions by adjusting

the parameter α. In the limit $\alpha \to \infty$, it is approximately the SE kernel (equation 10). Thus, the RQ kernel is more flexible than the SE kernel. If the modeler wishes the function to exhibit variations at multiple length scales, the RQ kernel is more suitable than the SE kernel.

Finally, we review the Matérn kernel defined by

$$k_{\text{Matérn}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu\, \mathbf{r}^\intercal \boldsymbol{\Lambda}^{-1} \mathbf{r}} \right)^{\nu} K_\nu \left(\sqrt{2\nu\, \mathbf{r}^\intercal \boldsymbol{\Lambda}^{-1} \mathbf{r}} \right). \tag{12}$$

Here, ν is a smoothness parameter, $\Gamma(\cdot)$ is the Gamma function, and $K_\nu(\cdot)$ is the modified Bessel function of the second kind. Like the RQ kernel (equation 11), Matérn kernels are a generalization of the SE kernel. It can be shown that in the limit $\nu \to \infty$, the Matérn kernel becomes the SE kernel[73]. Moreover, the SE kernel assumes infinitely differentiable (smooth) functions, while the Matérn kernel allows for varying degrees of smoothness through ν. These kernels can be useful when modeling real-world phenomena with unknown or varying smoothness, thereby providing more flexibility. Common choices for ν in machine learning and GP regresison applications literature include 1/2, 3/2 and 5/2[73]

In principled inference, the structure of the length scale matrix $\boldsymbol{\Lambda}$ is used to model (lack of) prior information about the function. In an isotropic GP, a single length scale is used for all input dimensions. Mathematically, this means the length scale matrix is written as $\boldsymbol{\Lambda} = \lambda^2 \mathbf{I}$. This modeling choice enforces that all input dimensions are equally important and have the same effect on the output. Alternatively, if one wanted to use separate length scales for each input dimension, one could select kernels (equations 10 - 12) with automatic relevance detection (ARD). This allows the kernel to capture the varying relevance of different dimensions, meaning that some dimensions can be more influential than others in predicting the output. Mathematically, this means the length scale matrix is written as $\boldsymbol{\Lambda} = \text{diag}(\lambda_1^2, \ldots, \lambda_d^2)$.

2.3.3 Gaussian Processes for Regression

Consider the regression setting in which a modeler is supplied with a dataset $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ composed of n pairs of regressors $\mathbf{x}_i \in \mathbb{R}^d$ and observations $y_i \in \mathbb{R}$. The goal is to recover the latent data generating process $f(\cdot)$. In most practical settings, the underlying process is perturbed by noise ε. That is,

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad i \in \{1, \ldots, n\},$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathscr{N}(0, \sigma_n^2)$. In GP regression (GPR) it is assumed that $f(\cdot) \sim \mathscr{GP}(m(\cdot), k(\cdot,\cdot))$. This assumption is called the prior. By linearity of expectation,

$$\mathbb{E}[y_i \,|\, \mathbf{x}_i] = m(\mathbf{x}_i)$$

and

$$\text{Cov}[y_i \,|\, \mathbf{x}_i, y_j \,|\, \mathbf{x}_j] = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 h_{i,j},$$

where $h_{i,j}$ is the Kronecker delta function

$$h_{i,j} = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

The goal in the regression setting is to predict $f(\cdot)$ over a test set $\mathbf{X}_* \in \mathbb{R}^{t \times d}$. Under the GP prior on $f(\cdot)$, the finite set of training and test outputs follows a joint multivariate normal distribution. That is,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 \mathbf{I} & \mathbf{K}_* \\ \mathbf{K}_*^\intercal & \mathbf{K}_{**} \end{bmatrix} \right).$$

Here, $\mathbf{K} \in \mathbb{R}^{n \times n}$, $\mathbf{K}_* \in \mathbb{R}^{n \times t}$, and $\mathbf{K}_{**} \in \mathbb{R}^{t \times t}$ are covariance matrices. To make predictions at the test points $\mathbf{X}_*$, one can leverage the conditional distribution of the test outputs given the training data $\mathscr{D}$. This is done with the finite-dimensional conditional distribution

$$\mathbf{f}_* \,|\, \mathbf{X}_*, \mathscr{D} \sim \mathscr{N}(\mathbf{K}_*^\intercal (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\mu}), \mathbf{K}_{**} - \mathbf{K}_*^\intercal (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}_*). \tag{13}$$

Note that this is the predictive distribution for $\mathbf{f}_*$. The predictive distribution for $\mathbf{y}_*$ can be obtained by adding $\sigma_n^2 \mathbf{I}$ to the covariance in equation 13.

2.3.4 Hyperparameter Estimation and Criteria for Model Selection

The behavior of mean and kernel functions is influenced by their parameters $\boldsymbol{\theta} = (\sigma_n, \sigma_f, \lambda_1, \ldots, \lambda_d)^\intercal$. If the elements of $\boldsymbol{\theta}$ are not chosen by the modeler, they must be inferred from the sample data $\mathscr{D}$. Furthermore, one might be interested in comparing the performance of several GP models and selecting the best performing model. The evidence (i.e., marginal likelihood) accomplishes both objectives.

The evidence is given by

$$p(\mathbf{y} \,|\, \mathbf{X}) = \int p(\mathbf{y} \,|\, \mathbf{f})\, p(\mathbf{f} \,|\, \mathbf{X})\, \mathrm{d}\mathbf{f},$$

where we marginalize over the function values $\mathbf{f}$. Given that both $p(\mathbf{y} \,|\, \mathbf{f})$ and $p(\mathbf{f} \,|\, \mathbf{X})$ are Gaussian, the marginal likelihood can be computed in closed form. Moreover, the marginal likelihood has a distribution

$$\mathbf{y} \,|\, \mathbf{X} \sim \mathscr{N}(\boldsymbol{\mu}, \mathbf{K} + \sigma_n^2 \mathbf{I}),$$

and the expression for the evidence is the probability distribution function of this distribution

$$p(\mathbf{y} \,|\, \mathbf{X}) = (2\pi)^{-n/2} |\mathbf{K} + \sigma_n^2 \mathbf{I}|^{-1/2} \exp\left(-\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})^\intercal (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\mu}) \right),$$

where $|\cdot|$ is the determinant. In practice, the negative log-marginal likelihood (LML) or log-evidence is minimized to find the optimal $\hat{\boldsymbol{\theta}}$, that is

$$\hat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}} \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\mathbf{K} + \sigma_n^2 \mathbf{I}| + \frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})^\intercal (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\mu}). \tag{14}$$

The terms in equation 14 aid in model selection as follows. The first component is the normalization constant, the second component is the model complexity penalty, and the third component is the model fit to the data. A smaller model fit term indicates better model fit. The determinant of the covariance matrix reflects the area or volume of the function space covered by the model. Thus, the larger (smaller) the determinant, the greater (lesser) the complexity of the model. Thus, equation 14 balances the trade-off between minimizing complexity and maximizing model fit.

2.3.5 GPs in the Context of this Work

Our goal is to develop GPR models that capture the trends shown in Figure 2. We postulate, based on Figure 1, that the JR GC predictions are a reasonable approximation for the experimental physical property measurements and thus assume a linear mean function equal to $\mathbf{y}_{\mathrm{GC}}$ with no additional trainable parameters. Thus, through this choice of mean function, our GPR models can be thought of as a hybrid model[81,82], where the GPR kernel corrects for the discrepancy between the JR GC prediction and the experimental data. We choose the rational quadratic (RQ) kernel with isotropic length scale parameter (no ARD) as the base kernel function for the GP models of every property to account for varying levels of smoothness. We add a white kernel with variance σ_w to the RQ kernel to account for uncertainty in the experimental data, such that the final covariance function used in this work is defined by equation 15.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left(1 + \frac{1}{2\alpha}\mathbf{r}^{\intercal}\boldsymbol{\Lambda}^{-1}\mathbf{r}\right)^{-\alpha} + \sigma_w^2 \delta_{i,j} \tag{15}$$

In the SI Section S1.3, we describe several alternate GPR model structures. For completeness (see Section 3.5), we compare these model alternatives but ultimately find the single model structure, described above, performs well for all six thermophysical properties and balances model performance with complexity. Thus, all of the results in the main text focus on this single model structure described above unless otherwise explicitly noted.

GP models were implemented using GPflow[94] (version 2.10.0). The hyperparameter tuning was implemented using scipy.optimize[95] (version: 1.13.1) with the Limited-memory Broyden–Fletcher–Goldfarb–Shanno Bound (L-BFGS-B) algorithm to perform maximum likelihood estimation and was repeated ten times to avoid local hyperparameter solutions. In the first training pass, all hyperparameters were initialized at 1.0. In subsequent repeats, ℓ and α were uniformly sampled from the bounds $[10^{-5}, 100]$. σ_f^2 was selected from a log-normal distribution with bounds $[0, 1.0]$ and σ_w was always initialized at 1.0. The optimization bounds for α were set to $[10^{-5}, 5\times10^{3}]$ and all other hyperparameters were optimized within the limits $[10^{-5}, 10^{2}]$. We checked the condition number of the kernel matrix $\mathbf{K}$ to ensure the GP models were reasonably scaled.

2.4 Stratified Sampling

When splitting the data into training and testing sets, an 80/20 split was used. In the final model implementation, all features and labels were standardized to have zero mean and unit variance using the scikit-learn StandardScaler[96]. Feature-based stratified sampling was used to split the data using an iterative stratification algorithm for multi-label data originally developed by Sechidis and co-workers[97] and further developed and implemented in the Scikit-Multilearn Python library by Szymański and Kajdanowicz[98]. A fixed random seed was used to ensure reproducibility of results across multiple training and retraining of the GPs in this work for all properties. The stratified sampling algorithm is robust to the choice of random seed (see SI), and for several of the properties, such as T_b, T_c, P_c, and V_c, using other random seeds did not change the stratified sampling train/test splits. Small changes in train/test splits and consequently in results were, however, observed for ΔH_{vap} and T_m when different random seeds were used. The results of using ten additional random seeds are summarized in SI Table S6 for ΔH_{vap} and T_m

2.5 Error Metrics

We used mean absolute error (MAE), mean absolute percent error (MAPE), coefficient of determination (R^2), and root mean squared error (RMSE) to quantify and analyze the prediction error of our GCGP models. We also computed the mean percentage error (MPE) for V_c predictions. Their definitions are as follows

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_{\mathrm{exp}_i} - \mu(\mathbf{x}_i)\right| \tag{16}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_{\mathrm{exp}_i} - \mu(\mathbf{x}_i)}{y_{\mathrm{exp}_i}}\right| \times 100\% \tag{17}$$

$$\mathrm{MPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mu(\mathbf{x}_i) - y_{\mathrm{exp}_i}}{y_{\mathrm{exp}_i}} \times 100\% \tag{18}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_{\mathrm{exp}_i} - \mu(\mathbf{x}_i)\right)^2}{\sum_{i=1}^{N}\left(y_{\mathrm{exp}_i} - \overline{y_{\mathrm{exp}}}\right)^2} \tag{19}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{exp}_i} - \mu(\mathbf{x}_i)\right)^2} \tag{20}$$

Note that in equation 19, $\overline{y_{\mathrm{exp}}}$ is the average value of $\mathbf{y}_{\mathrm{exp}}$.

3 Results and Discussion

3.1 GCGP Method Accurately Predicts Properties and Corrects Systematic Bias

In this work, we used a GP to correct for the systematic bias of the JR GC method. The results are presented in Figure 3 organized by the six thermophysical properties. SI Table S2 lists the optimized GP hyperparameters for each property.

The GCGP method provides significant correction to the systematic bias in the JR GC models (see Figure 3). The coefficient of determination (R^2) values of the predictions of the GCGP test set are ≥ 0.85 for five out of six and ≥ 0.90 for four out of six properties modeled in this work. The MAPE values of the testing set are less than 5.5% for five of the six properties modeled. These prediction accuracy metrics are competitive when compared to other ML-related efforts in the literature[79,99–103] to predict some of the properties modeled in this work. Some of these methods in the literature utilize tens to hundreds of input features[79,99,103], with

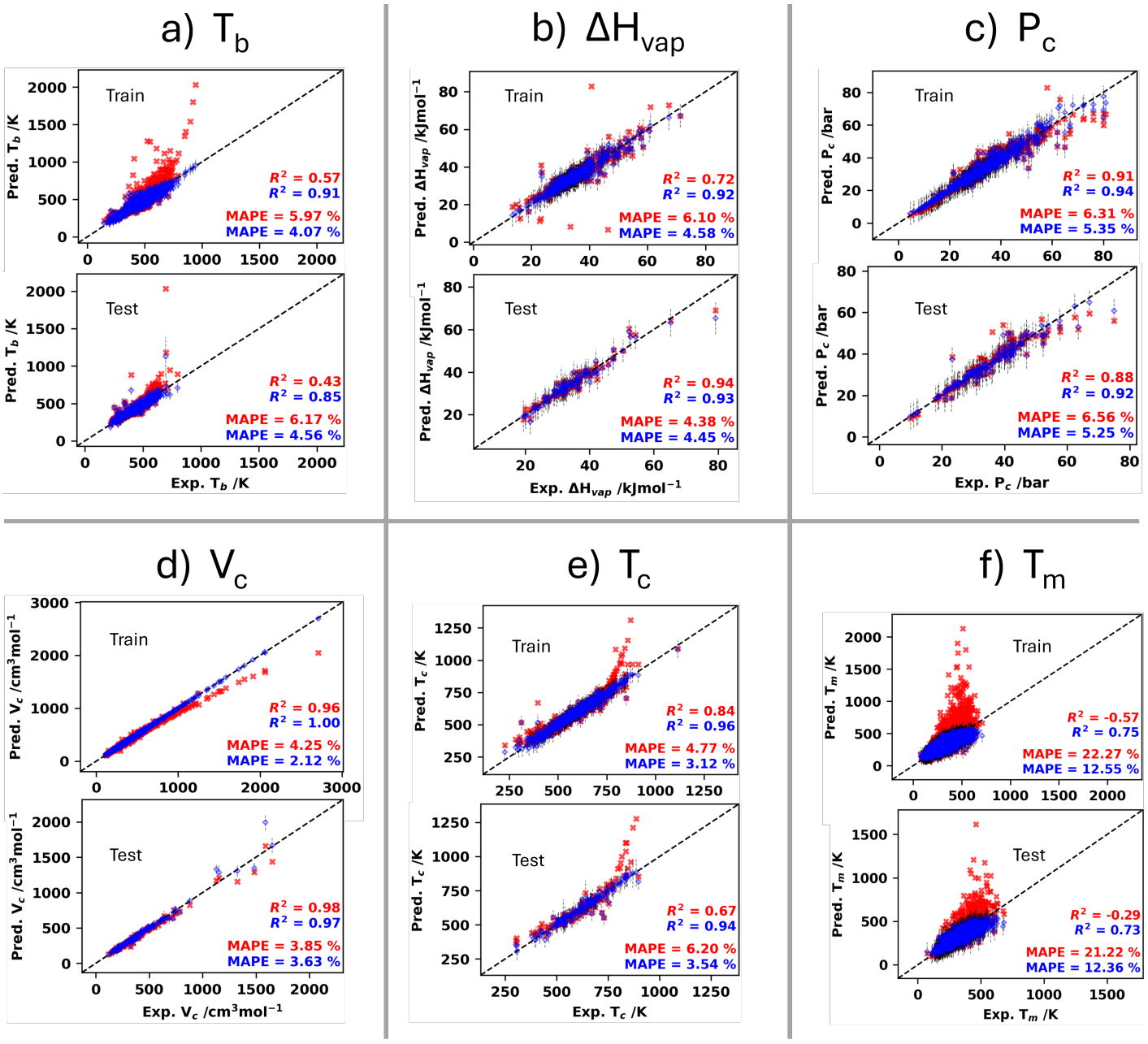

Fig. 3 GCGP corrections of systematic bias in JR GC model. Red are JR GC predictions, and blue are GCGP prediction means with predicted 95 % confidence intervals shown using black broken-line error bars.

some requiring quantum mechanical calculations of molecular descriptors [99,104] or energy minimization of molecular structures [100] to generate input features. The GCGP method uses only two input features derived from fast and straightforward GC-based calculations. Furthermore, the same input feature type is used for all properties, potentially eliminating the need to individually determine a unique set of input features for every material property prediction task, which is the current norm in the literature.

The GCGP method provides the greatest improvement for T_m, for which the JR GC method exhibits the greatest systematic bias (see Figure 1). Performance metrics for the original JR GC method for T_m are poor: test set $R^2 = -0.29$ and MAE = 75.0 K. In contrast, the proposed GCGP method is much more accurate for T_m with test set $R^2 = 0.73$ and MAE = 40.6 K. This is remarkable because the GP has only two input features: the molecular weight and the GC predictions, which often exhibit significant bias.

As discussed in Section 2.2.1, molecules with very different functional groups and molecular structures can have similar values of T_m. For example, the aliphatic hydrocarbon 2-butyne with molecular formula C_4H_6 and the aromatic compound N,N-dibutylaniline with molecular formula $C_{14}H_{23}N$ both have the same T_m value of 240.95 K according to the CRC Handbook of Chemistry and Physics [18]. These values agree with the values reported in the NIST webbook [91]. This convoluted or unclear link

between molecular constitution and structure with T_m makes it difficult for the GP to learn and correct the systematic bias in the JR GC predictions for T_m. This may also explain why the JR GC method performs extremely poorly for T_m prediction. Other works in the literature[87,104,105] have encountered similar challenges in using ML techniques for the prediction of T_m. Hughes et al.[104] reported that T_m was the most difficult property to predict among the several properties they considered in their work. Hughes et al. used 168 2D and 53 3D (221 total) molecular descriptors obtained from quantum mechanical calculations. The best testing set R^2 obtained for T_m in their work was 0.46[104]. Li et al.[87] used deep learning with protein sequences as input features for predicting T_m for proteins and obtained a testing set R^2 of 0.75 for T_m. Venkatraman et al.[105] used several ML techniques using semi-empirical (PM6) electronic, thermodynamic, and geometrical descriptors to predict T_m for ionic liquids. The testing set R^2 values ranged from 0.53 to 0.67 for different ML techniques. The GCGP T_m predictive performance is thus competitively comparable to other (more complicated) methods in the literature for T_m, potentially offering better predictive performance while maintaining computational efficiency and parsimoniousness. Table S3 in the SI shows the effect of different settings of the white noise kernel variance (σ_w^2) on the model training metrics for T_m in our work. The JR GC method also shows significant systematic bias for T_b and T_c. The application of the GCGP method significantly increased the testing set R^2 values from 0.43 to 0.85 and from 0.67 to 0.94 for T_b and T_c, respectively. The results for T_m and T_b show that the GCGP method greatly improves the predictive accuracy of simple GC-based models, especially for scenarios where the GC models have extremely poor predictive performance.

The GCGP method also provides correction to observable systematic bias even when the systematic bias is small, and the overall predictive accuracy of the JR GC method is very high. The results for V_c in Figure 3 demonstrate this. The testing set R^2 for the JR GC prediction of V_c is 0.98. The GCGP method did not increase the R^2 for the testing set, and it may thus seem that there was no correction of bias gained by applying the GCGP method. The MPE value for the GC prediction of V_c for the test set is -1.54%, while the GCGP MPE for the test set is -0.08%. A comparison of the MPE for the predictions of V_c, coupled with visual observation of V_c results in Figure 3, allows us to infer that the systematic underestimation of V_c for molecules with higher MW and V_c in the GC predictions was corrected. The prediction error became no longer observably systematically biased using the GCGP method. A significantly negative MPE indicates systematic underestimation, as is the case for the GC-only predictions. This is in agreement with the observed V_c results in Figure 3 for both the training and testing set results.

Overall, the GCGP method offers a novel approach for accurately and efficiently predicting thermophysical properties, is applicable to a wide range of properties, and utilizes a significantly lower number of input features compared to most of the other predictive ML-based models in the literature. Section 3.3 provides a discussion of the ΔH_{vap} and P_c results.

3.2 GCGP is Significantly more Accurate than JR Methods Alone

For every thermophysical property, the MAE (equation 16) and RMSE (equation 20) were assessed for both the JR model and GCGP models. To assess model performance, these metrics were compared across training and testing datasets. Figure 4 summarizes these findings.

Figure 4 shows the GCGP models are more accurate than the JR GC predictions for all of the properties. The only exception appears to be that the test error metrics in the JR GC model (Figure 4 (a)) are marginally less than those of the GCGP model for ΔH_{vap} (Figure 4 (b)) as also observed in Figure 3. This is due to the nuanced bias in JR GC ΔH_{vap} predictions, which is discussed in more detail in Section 3.3.

Figure 4 (b) shows that for all models, there is more error in the test set than in the training set. This trend is reasonable, as one would expect to see slightly more error in out-of-sample predictions. The only exception is the RMSE value for P_c. The training set RMSE for GCGP P_c prediction is marginally higher than the testing set value. We consider this to be an artifact of the train/test split. Furthermore, the difference in RMSE values between training and testing sets for P_c is almost identical.

The stratified sampling method used in this work is robust to the choice of random seed in train/test splits; however, for T_m and ΔH_{vap}, different random seeds give slightly different train/test splits. SI Table S6 shows that for most random seed choices, the training performance metrics are better than the testing performance metrics, and the training set errors are generally lower than those of the testing set, as expected. Furthermore, SI Table S6 shows that for all random seed choices, the model performance metrics for the testing set are not widely different, indicating that the GCGP method is robust to the choice of train/test splits.

3.3 GCGP Corrects Nuanced Systematic Bias for ΔH_{vap} and Provides Accurate Out-of-Sample Predictions

The systematic bias for ΔH_{vap} is subtle. For most molecules, the bias in JR GC ΔH_{vap} predictions is small and thus the GCGP method provides negligible improvement in predictive accuracy (see Section 3.2 and Figure 4). However, the systematic bias for highly fluorinated (12 atoms or more) or highly nitrated molecules is large. The GCGP method provides the greatest improvement in predictive accuracy for these molecules with the greatest systematic bias. This is shown in Figure 5 (numerical values are presented in SI Table S4) for the case of highly fluorinated molecules.

There were only two highly fluorinated molecules and one highly nitrated molecule in the collected experimental data for ΔH_{vap}. The highly fluorinated molecules had the lowest JR GC ΔH_{vap} predictions, and the highly nitrated molecule had the highest JR GC ΔH_{vap} predictions in our data set (see Figure 1). Our use of stratified sampling based on the input features ensured that the data for these three molecules were placed in the training set. To demonstrate that the GCGP method indeed learned and was able to correct for the unique chemical constituent-based sys-

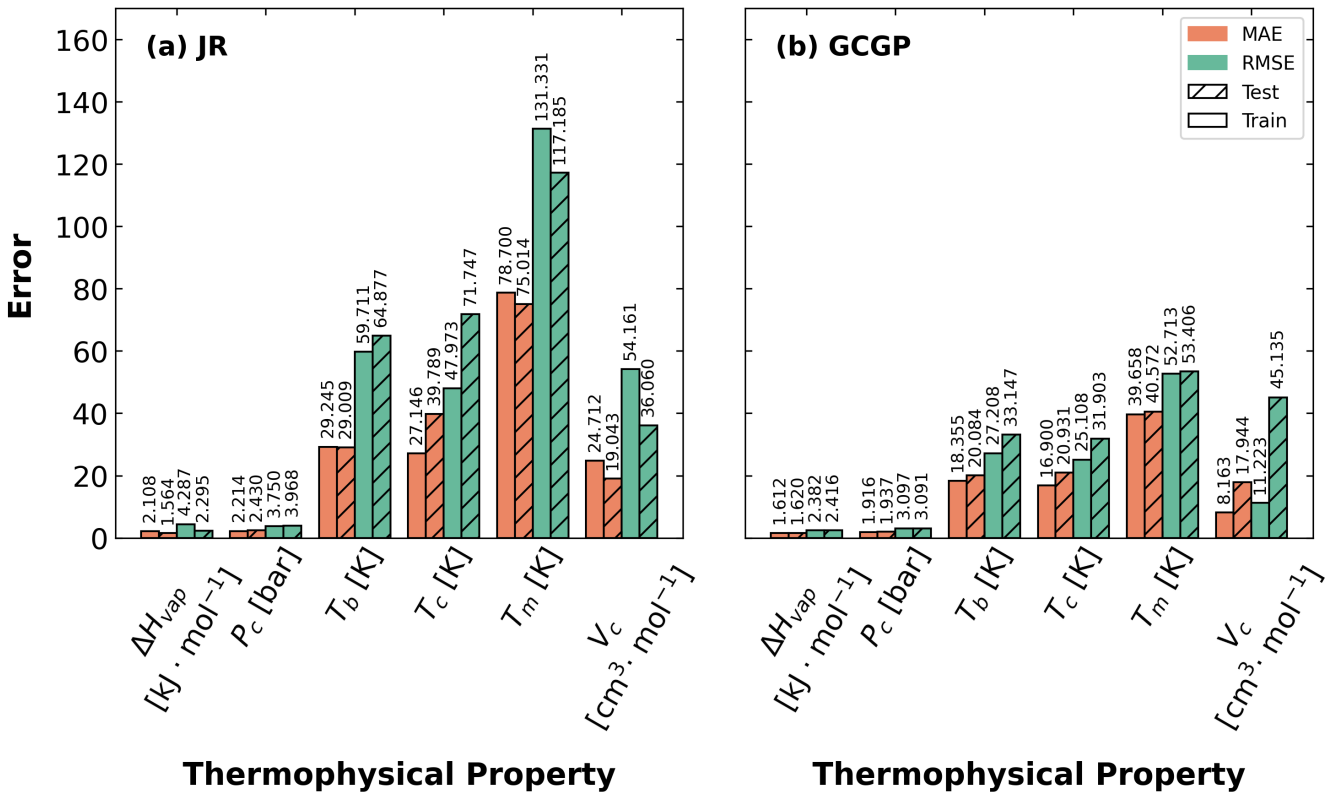

Fig. 4 Error vs. thermophysical property of selected GP models for (a) JR and (b) GC-GP models. Salmon (green) represents MAE (RMSE). Solid colors (stripes) represent the training (testing) data.

tematic bias for ΔH_{vap}, we obtained additional experimental data for five highly fluorinated molecules from Yaws' Critical Property Data for Chemical Engineers and Chemists as available in the Knovel database. We obtained JR GC ΔH_{vap} predictions for these molecules and then applied the GCGP method to also predict ΔH_{vap} for the molecules with GCGP predicted uncertainties (See Figure 5).

Figure 5 shows how well the GCGP method corrects systematic bias and significantly improves the accuracy of ΔH_{vap} predictions for highly fluorinated molecules. None of the five molecules in Figure 5 were present in the original ΔH_{vap} data set (both training and testing) used in this work. No highly fluorinated molecules were in the testing set in the original data set, as the two highly fluorinated molecules in the original data set were placed in the training set by the stratified sampling method. Interestingly, the GP leveraged sparse training data from the region of the input feature space corresponding to highly fluorinated molecules and was able to correct the systematic bias in JR GC ΔH_{vap} predictions with high accuracy. This further underscores the power of the GCGP method. Similar results can be expected for ΔH_{vap} predictions for highly nitrated compounds and for P_c predictions for highly brominated compounds.

This result is notable when considering that, unlike most ML methods in the literature, which utilize input features that encode the chemical identity of molecules in detail, our approach does not explicitly provide the chemical identity of molecules to the GPs. Our GPs are not explicitly informed about the presence or absence of certain chemical moieties, yet they perform well in correcting systematic bias that arises from the presence and quantity of these chemical moieties in molecules.

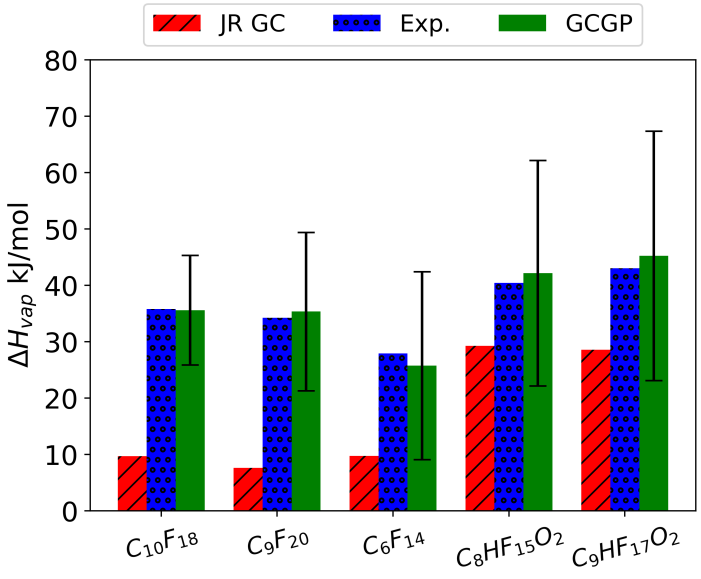

Fig. 5 Comparison of GCGP and JR GC ΔH_{vap} predictions for five highly fluorinated molecules not in the training data set. Error bars visualize 95 % prediction intervals. Experimental data from Yaws' Critical Property Data for Chemical Engineers and Chemists[17]

3.4 GCGP 95% prediction intervals are reliable for unseen data

Importantly, the GCGP method provides uncertainty estimates that are usually not available from GC methods. We now analyze the reliability of GCGP 95% prediction intervals.

Figure 6 shows the percentage of GCGP 95% prediction intervals that overlap with the experimental values for both the training and testing sets. We observe that for the training sets for all properties, the percentage of data points whose 95% prediction intervals overlap with the experimental data points is greater than 95%, with P_c and T_m being the only exceptions with 93.42% and 94.35%, respectively.

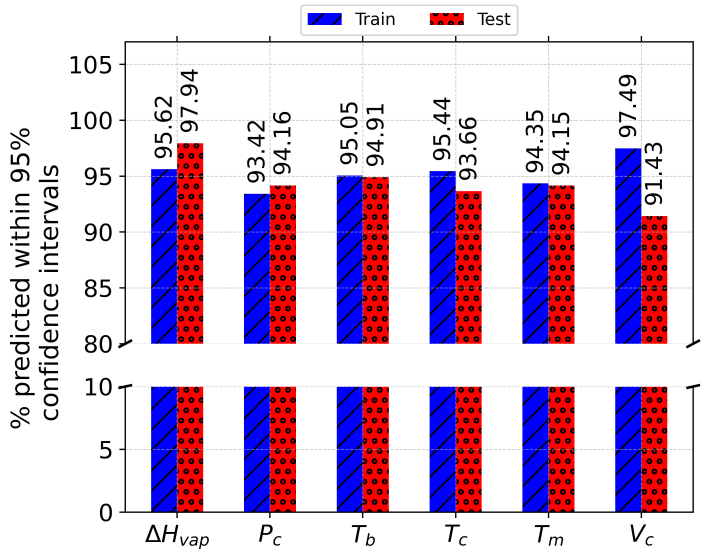

Fig. 6 Percentage of GCGP predictions that match with experimental data within predicted 95% confidence interval

A more interesting analysis is how well the prediction intervals overlap with the experimental values for 'unseen' data (testing set). Remarkably, for all six properties, the percentage of the testing set predictions with 95% prediction intervals overlapping with the experimental values is greater than 90% and greater than 94% for four of the six properties modeled.

GP predicted uncertainties for ΔH_{vap} for highly fluorinated molecules (out-of-sample data) are shown in SI Table S4 with 95% prediction intervals visualized as error bars in Figure 5. These predicted uncertainties are higher than the average uncertainties in the training and testing set predictions for the original ΔH_{vap} data. The high uncertainties are expected due to the sparsity of data in the input feature space corresponding to highly fluorinated molecules in the training dataset.

Therefore, the 95% prediction intervals from the GCGP method are reliable for unseen or new molecules and can be reasonably expected to have a greater than 90% empirical likelihood of representing the range of the true values even in the absence of experimental data. This is particularly important when screening new molecules for a range of applications using the GCGP method.

3.5 GCGP approach is robust across kernel and model structure choices

For completeness, we now consider different GCGP model design choices, including kernel selection, ARD application, and the overall model structure. The complete results of the assessment of the sensitivity of the GCGP method to kernel design and model structure are archived in the companion GitHub repository.

In assessing the sensitivity of the GCGP method to kernel design and model structure, we will focus our discussion on the LML defined in Equation 14. Figures 7 (a)-(f) show the LML for each thermophysical property investigated. For the four model architectures investigated, five isotropic parameterizations of different kernel functions were assessed. The LML of the anisotropic RQ kernel is also shown to allow comparison between the anisotropic and isotropic kernels for the six thermophysical properties studied.

The descriptions of the four model structures tested are provided in the SI subsections S1.3 with equations S1 - S4 in the SI representing Models 1 through 4, respectively. Model 3 is the final model implemented in this work.

We note that the formulation of the LML does not explicitly and fully account for model complexity that may arise due to differences in the number of parameters in the mean function, especially for low-data scenarios, as we have in this work.

We have applied information from computed LML values, keeping in mind the limitation highlighted above. Uncertainties in computed LML values may arise from randomness in train/test splits, randomness in kernel hyperparameter initialization during retrainings, uncertainties in the optimized hyperparameters, and other factors. In the following discussions, LML values within a 1% difference or an absolute LML difference of 1.0 from each other (whichever is greater) are considered similar. More details are provided in the SI subsection S2.4.1.

For the RQ kernel, we find that the LML values for anisotropic kernels are similar to those for isotropic kernels for all properties except ΔH_{vap} as shown in Figure 7. Similar results are observed for all other anisotropic kernels, compared to their isotropic counterparts, regardless of the kernel functional form. Based on these results, we chose to implement the final model using isotropic kernels for all properties.

Also, Figure 7 shows that Model 2 (eq.S2) performs the worst for all thermophysical properties. This result is as expected, as Model 2 is not complex enough to be informative. Furthermore, Model 2 is the only model that utilizes a single descriptor (molecular weight). Thus, molecular weight alone is not a good enough descriptor to model GC discrepancy. Taken as a whole, these results justify our decision to include both molecular weight and GC prediction as descriptors.

Model 3 was found to give slightly better LML values compared to Model 1 overall. Model 3 has a more physically meaningful and intuitive mean function with no additional trainable parameters compared to Model 1. Model 1, however, performed better than Model 3 for properties that had very poor GC predictions, such as T_m. This is expected since the use of the JR GC predictions as the mean function is less valid when the GC predictions are poor.

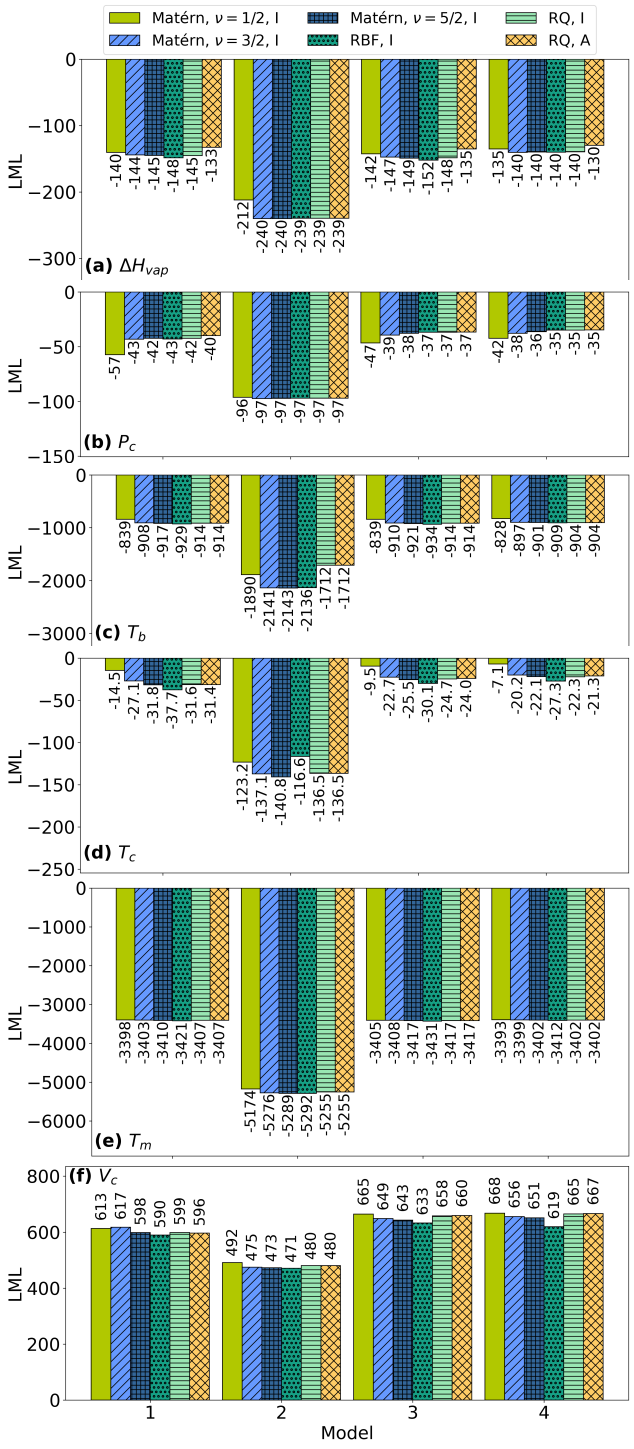

Fig. 7 LML (eq. (14)) vs. model architecture 1-4 (eqs. SI S1 - SI S4). I = isotropic kernel, A = anisotropic kernel. (a) heat of vaporization, ΔH_{vap}, (b) critical pressure, P_c, (c) boiling temperature, T_b, (d) critical temperature, T_c, (e) melting temperature, T_m, and (f) critical volume, V_c.

Model 4, with almost double the number of trainable parameters, with three additional parameters compared to other models, had similar LML values compared to Model 3 for T_m and V_c. Model 4 had slightly better LML values compared to Model 3 for other properties. Considering the significantly higher number of additional trainable parameters in Model 4, while offering only a slight improvement in LML values compared to Model 3 in general, we chose Model 3 for the final model implementations.

Finally, we find that given the selection of Model 3 and isotropic kernels for final model implementation, the RQ kernel with an additional trainable parameter known as the shape parameter α, has more flexibility to model the range of properties studied in this work, regardless of the smoothness (or roughness) of the surface to be learned. Further discussion is provided in the SI subsection S2.4.2 and the kernel choice rankings in the SI Table S5.

Regardless of kernel choice, anisotropy settings, or model structure (with the exception of Model 2), the GCGP method generally gives good and comparable predictive performance. Therefore, the GCGP method is robust to kernel choice and design and also robust to model structure, with the exception of overly simplistic modeling choices like Model 2.

4 Conclusions

We have developed and demonstrated a material property prediction method that integrates the strengths of GC-based molecular models for property predictions with GPR to improve prediction accuracy and provide reliable uncertainty estimates. The GCGP method corrects systematic bias in GC-based property modeling and offers significant improvement in predictive accuracy over GC-only predictions. The GCGP method can correct nuanced systematic bias associated with the presence of specific structural units in molecules, even though the GPs are not explicitly exposed to information about the presence and amounts of these structural units. The GCGP approach is robust to the choice of GP kernels and model structure, provided the GC predictions are used as one of the input features to the GP. Furthermore, the GCGP method has great potential to give even better predictive accuracies through proper tuning. It can be conveniently extended to other properties, GC models, and molecule types not considered in this work. The GCGP method developed in this work thus offers a fast, simple, reliable, generalizable, and tunable property prediction method that gives predicted uncertainties for the property predictions. Although this work focuses on six properties and the JR GC method, the technique for creating models is directly applicable to other properties and other GC methods. The GCGP method, therefore, offers a key tool for reliable property prediction for material screening in material discovery tasks.

We conclude by highlighting some limitations and opportunities for future research related to the development and application of the proposed GCGP method.

The first limitation of the GCGP method that we highlight is that its performance may be limited by the accuracy of the input GC method, as is the case for T_m in this work. This provides an opportunity for tunability for improved predictive accuracy for properties that are difficult to predict, such as T_m. A simple way

to improve the prediction of T_m, for example, using the GCGP approach, is to switch to a more accurate but still simple GC method for predicting GC T_m. In fact, such a GC method already exists [106]. Alternatively, we can use the same structural unit definition as the JR GC method, but design and parameterize a more accurate GC model functional form to provide a more accurate input for the GCGP method.

A second limitation of the GCGP method is that its use requires that the molecule for which property predictions are required must be treatable using a specific GC method. One way to overcome this limitation for molecules that cannot have their properties predicted due to the limitations of unavailable parameters in a given GC method is to have their properties predicted by switching the GC method to another one that is able to predict their properties. This may entail developing a multi-GCGP method that is capable of receiving GC prediction inputs from multiple GC methods to help mitigate the limitation of the inability of an individual GC method to cover all of chemical space. For this to work successfully, the identity of the GC method providing prediction input for a given molecule has to be encoded and provided as an additional input feature to the GP. A simpler but less elegant solution may be to build multiple separate GCGP models for the same property, each covering some area of chemical space that other GC methods may not cover.

A third limitation of the GCGP method is that its ability to handle the property prediction of isomers is limited to the underlying GC method's ability to distinguish between isomers. Higher order GC methods have been developed to help mitigate some of the challenges with property prediction involving isomers using GC methods [45,46]. An interesting opportunity will be to incorporate low-dimensional topological indices such as the Weiner index [107], the Zagreb indices [108], and Randic index [109] as additional inputs to the GP. This will have a drawback of higher input feature space dimensions, but can potentially greatly improve the differentiability of isomers for property prediction using the GCGP method.

Furthermore, one opportunity for future contributions will be to extend the GCGP method to predict properties under varying conditions of temperature and possibly pressure. This may be achieved by adding temperature as an input feature to the GP and training against sufficient data to capture the temperature dependence of the target property.

Finally, a contribution that would be highly valuable is integrating the GCGP method with CAMD workflows. The improved predictive accuracies and easily accessible and reliable uncertainty estimates from the GCGP method could result in a significant improvement in the reliability and robustness of CAMD workflows in the identification of optimal molecules and processes for a variety of applications. [110]

Acknowledgements

The authors acknowledge funding from the National Science Foundation (NSF) EFRI DChem: Next-generation Low Global Warming Refrigerants, Award no. 2029354. EJM, AD, MC, and BA acknowledge that this research is based upon work supported by the National Science Foundation under award number ERC-2330175 for the Engineering Research Center EARTH. DA acknowledges funds from the projects CICECO-Aveiro Institute of Materials, UIDP/50011/2020 (DOI 10.54499/UIDP/50011/2020) and LA/P/0006/2020 (DOI 10.54499/LA/P/0006/2020), financed by Portugal's national funds through the FCT/MCTES (PIDDAC). BA & KJ acknowledge the Notre Dame Lucy Family Institute for Data and Society. BA acknowledges the Center for Sustainable Energy at Notre Dame, for graduate research fellowship. MC & KJ acknowledge support from the Graduate Assistance in Areas of National Need fellowship from the Department of Education via grant number P200A210048, the National Science Foundation via Award numbers CBET-1917474 and EFRI-2029354, and the University of Notre Dame College of Engineering and Graduate School. Computational resources were provided by the Center for Research Computing (CRC) at the University of Notre Dame.

Notes and references

1 J. Bhattacharjee and S. Roy, *Mat. Sci. Res. India*, 2023, **20**, 141–145.
2 M. McGrath, *Climate Change: 'Monumental' Deal to Cut HFCs, Fastest Growing Greenhouse Gases*, 2016, `https://www.bbc.com/news/science-environment-37665529`.
3 G. A. Ozin and J. Y. Y. Loh, *Energy Materials Discovery: Enabling a Sustainable Future*, Royal Society of Chemistry, 2022.
4 D. A. Giannakoudakis, L. Meili and I. Anastopoulos, *Novel Materials for Environmental Remediation Applications: Adsorption and Beyond*, Elsevier, 2022.
5 K. C. Nicolaou, *Angewandte Chemie*, 2014, **126**, 9280–9292.
6 C. Davenport, *Nations, Fighting Powerful Refrigerant That Warms Planet, Reach Landmark Deal*, 2016, `https://www.nytimes.com/2016/10/15/world/africa/kigali-deal-hfc-air-conditioners.html`.
7 Department of Ecology, State of Washington, *Hydrofluorocarbons*, https://ecology.wa.gov/Air-Climate/Reducing-Emissions/Hydrofluorocarbons, 2023.
8 United States Environmental Protection Agency, *Reducing Hydrofluorocarbon (HFC) Use and Emissions in the Federal Sector through SNAP*, https://www.epa.gov/snap/reducing-hydrofluorocarbon-hfc-use-and-emissions-federal-sector-through-snap, 2014.
9 M. O. McLinden and M. L. Huber, *J. Chem. Eng. Data*, 2020, **65**, 4176–4193.
10 N. Wang, M. N. Carlozo, E. Marin-Rimoldi, B. J. Befort, A. W. Dowling and E. J. Maginn, *J. Chem. Theory Comput.*, 2023, **19**, 4546–4558.
11 R. W. Smith and E. J. Maginn, *Mol. Simul.*, 2024, **50**, 26–42.
12 E. Marin-Rimoldi, A. D. Yancey, M. B. Shiflett and E. J. Maginn, *J. Chem. Phys.*, 2024, **161**, 074701.
13 K. R. Baca, K. Al-Barghouti, N. Wang, M. G. Bennett, L. Matamoros Valenciano, T. L. May, I. V. Xu, M. Cordry, D. M. Haggard, A. G. Haas, A. Heimann, A. N. Harders, H. G. Uhl, D. T. Melfi, A. D. Yancey, R. Kore, E. J. Maginn, A. M. Scurto and M. B. Shiflett, *Chem. Rev.*, 2024, **124**, 5167–5226.
14 B. Agbodekhe, E. Marin-Rimoldi, Y. Zhang, A. W. Dowling

and E. J. Maginn, *J. Chem. Eng. Data*, 2024, **69**, 427–444.
15 C. K. Z. Andrade and L. M. Alves, *Curr. Org. Chem.*, 2005, **9**, 195–218.
16 S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang and E. E. Bolton, *Nucleic Acids Res.*, 2023, **51**, D1373–D1380.
17 C. L. Yaws, *Yaws' Critical Property Data for Chemical Engineers and Chemists*, Knovel, 2014.
18 J. R. Rumble, *CRC Handbook of Chemistry and Physics*, CRC Press/Taylor & Francis, Boca Raton, FL, 105th edn, 2023.
19 *Dortmund Data Bank*, `http://www.ddbst.com`, 2024, Accessed: 2024.
20 M. O. McLinden, J. S. Brown, R. Brignoli, A. F. Kazakov and P. A. Domanski, *Nat. Commun.*, 2017, **8**, 14476.
21 C. S. Adjiman, N. V. Sahinidis, D. G. Vlachos, B. Bakshi, C. T. Maravelias and C. Georgakis, *Ind. Eng. Chem. Res.*, 2021, **60**, 5194–5206.
22 F. Gharagheizi, P. Ilani-Kashkouli and A. H. Mohammadi, *Chem. Eng. Sci.*, 2012, **78**, 204–208.
23 Á. K. S. S. C. Chagas, A. L. H. Costa, P. H. R. Alijó and E. R. A. Lima, *Chem. Eng. Sci.*, 2021, **244**, 116796.
24 S.-K. Oh and K.-H. Park, *Korean J. Chem. Eng.*, 2005, **22**, 268–275.
25 R. L. Gardas and J. A. P. Coutinho, *AIChE J.*, 2009, **55**, 1274–1290.
26 S.-K. Oh and S. W. Campbell, *Fluid Phase Equilib.*, 1997, **129**, 69–88.
27 K. Nasrifar and M. Moshfeghian, *Fluid Phase Equilib.*, 1998, **153**, 231–242.
28 J. Li, M. Topphoff, K. Fischer and J. Gmehling, *Ind. Eng. Chem. Res.*, 2001, **40**, 3703–3710.
29 S. Tamouza, J. P. Passarello, P. Tobaly and J. C. de Hemptinne, *Fluid Phase Equilib.*, 2005, **228–229**, 409–419.
30 D. Nguyenhuynh, J. P. Passarello, P. Tobaly and J. C. de Hemptinne, *Fluid Phase Equilib.*, 2008, **264**, 62–75.
31 T. X. Nguyen-Thi, S. Tamouza, P. Tobaly, J.-P. Passarello and J.-C. de Hemptinne, *Fluid Phase Equilib.*, 2005, **238**, 254–261.
32 S. Dufal, V. Papaioannou, M. Sadeqzadeh, T. Pogiatzis, A. Chremos, C. S. Adjiman, G. Jackson and A. Galindo, *J. Chem. Eng. Data*, 2014, **59**, 3272–3288.
33 A. J. Haslam, A. González-Pérez, S. Di Lecce, S. H. Khalit, F. A. Perdomo, S. Kournopoulos, M. Kohns, T. Lindeboom, M. Wehbe, S. Febra, G. Jackson, C. S. Adjiman and A. Galindo, *J. Chem. Eng. Data*, 2020, **65**, 5862–5890.
34 M. Fayaz-Torshizi and E. A. Müller, *Macromol. Theory Simul.*, 2022, **31**, 2100031.
35 Å. Ervik, A. Mejía and E. A. Müller, *J. Chem. Inf. Model.*, 2016, **56**, 1609–1614.
36 G. M. Kontogeorgis and G. K. Folas, *Thermodynamic Models for Industrial Applications: From Classical and Advanced Mixing Rules to Association Theories*, John Wiley & Sons, Ltd, Chichester, UK, 2009.
37 A. Fredenslund, *Vapor-Liquid Equilibria Using Unifac: A Group-Contribution Method*, Elsevier, 2012.
38 M. T. White, O. A. Oyewunmi, A. J. Haslam and C. N. Markides, *Energy Convers. Manag.*, 2017, **150**, 851–869.
39 M. Lampe, M. Stavrou, J. Schilling, E. Sauer, J. Gross and A. Bardow, *Comput. Chem. Eng.*, 2015, **81**, 278–287.
40 M. Lampe, C. Kirmse, E. Sauer, M. Stavrou, J. Gross and A. Bardow, *Computer Aided Chemical Engineering*, Elsevier, 2014, vol. 34, pp. 357–362.
41 N. G. Chemmangattuvalappil, *Curr. Opin. Chem. Eng.*, 2020, **27**, 51–59.
42 P. J. Walker, H.-W. Yew and A. Riedemann, *Ind. Eng. Chem. Res.*, 2022, **61**, 7130–7153.
43 K. Joback and R. Reid, *Chem. Eng. Commun.*, 1987, **57**, 233–243.
44 A. L. Lydersen, *Estimation of Critical Properties of Organic Compounds by the Method of Group Contributions. Engineering Experiment Station Report 3. College of Engineering, University of Wisconsin, Madison, Wisconsin*, 1955.
45 J. Marrero and R. Gani, *Fluid Phase Equilib.*, 2001, **183–184**, 183–208.
46 L. Constantinou and R. Gani, *AIChE J.*, 1994, **40**, 1697–1710.
47 S.-T. Le, T. C. G. Kibbey, K. P. Weber, W. C. Glamore and D. M. O'Carroll, *Sci. Total Environ.*, 2021, **764**, 142882.
48 R. Al, J. Frutiger, A. Zubov and G. Sin, *Computer Aided Chemical Engineering*, Elsevier, 2018, vol. 44, pp. 1723–1728.
49 T. A. Albahri, *Chem. Eng. Sci.*, 2003, **58**, 3629–3641.
50 J. Frutiger, C. Marcarie, J. Abildskov and G. Sin, *J. Hazard. Mater.*, 2016, **318**, 783–793.
51 F. Gharagheizi, *J. Hazard. Mater.*, 2009, **170**, 595–604.
52 R. N. Walters and R. E. Lyon, *J. Appl. Polym. Sci.*, 2003, **87**, 548–563.
53 G.-B. Wang, C.-C. Chen, H.-J. Liaw and Y.-J. Tsai, *Ind. Eng. Chem. Res.*, 2011, **50**, 12790–12796.
54 F. Jirasek, N. Hayer, R. Abbas, B. Schmid and H. Hasse, *Phys. Chem. Chem. Phys.*, 2023, **25**, 1054–1062.
55 M. D. Wessel and P. C. Jurs, *J. Chem. Inf. Comput. Sci.*, 1995, **35**, 68–76.
56 J. C. Dearden, *Environ. Toxicol. Chem.*, 2003, **22**, 1696–1709.
57 J. Taskinen and J. Yliruusi, *Adv. Drug Deliv. Rev.*, 2003, **55**, 1163–1183.
58 C. Y. Zhao, H. X. Zhang, X. Y. Zhang, M. C. Liu, Z. D. Hu and B. T. Fan, *Toxicology*, 2006, **217**, 105–119.
59 C. X. Xue, R. S. Zhang, H. X. Liu, M. C. Liu, Z. D. Hu and B. T. Fan, *J. Chem. Inf. Comput. Sci.*, 2004, **44**, 1267–1274.
60 O. Obrezanova, G. Csányi, J. M. R. Gola and M. D. Segall, *J. Chem. Inf. Model.*, 2007, **47**, 1847–1857.
61 I. Pustokhina, A. Seraj, H. Hafsan, S. M. Mostafavi and S. M. Alizadeh, *Int. J. Chem. Eng.*, 2021, **2021**, 5650499.
62 S. Bishnoi, R. Ravinder, S. H. Grover, H. Kodamana and N. M. Anoop Krishnan, *Mater. Adv.*, 2021, **2**, 477–487.
63 D. S. Palmer, N. M. O'Boyle, R. C. Glen and J. B. O. Mitchell,

J. Chem. Inf. Model., 2007, **47**, 150–158.
64 C. P. Gupta, V. C. Srivastava and A. Dvivedi, *Eng. Appl. Artif. Intell.*, 2025, **157**, 111328.
65 A. H. Milyani, M. Karimi, A. Alizadeh, N. Nasajpour-Esfahani, N. H. Abu-Hamdeh, M. Hekmatifar and M. Shamsborhan, *J. Mol. Liq.*, 2023, **387**, 122625.
66 E. B. Postnikov, B. Jasiok and M. Chorążewski, *J. Mol. Liq.*, 2021, **333**, 115889.
67 S. A. Tawfik, O. Isayev, M. J. S. Spencer and D. A. Winkler, *Adv. Theory Simul.*, 2020, **3**, 1900208.
68 A. Nandy, C. Duan and H. J. Kulik, *Curr. Opin. Chem. Eng.*, 2022, **36**, 100778.
69 Y. Liu, Z. Yang, X. Zou, S. Ma, D. Liu, M. Avdeev and S. Shi, *Natl. Sci. Rev.*, 2023, **10**, nwad125.
70 J. Schmidt, M. R. G. Marques, S. Botti and M. A. L. Marques, *npj Comput. Mater.*, 2019, **5**, 83.
71 K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, *Nature*, 2018, **559**, 547–555.
72 M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov and S. Nahavandi, *Inf. Fusion*, 2021, **76**, 243–297.
73 C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
74 V. Villazón-León, R. R. Suárez, A. Bonilla-Petriciolet and J. C. Tapia-Picazo, *Fluid Phase Equilib.*, 2025, **595**, 114395.
75 S. Ma, L. Yang, C. Yumin, L. Xinyan and Y. Chen, *Chem. Eng. Commun.*, 2025, **212**, 1213–1232.
76 A. R. N. Aouichaoui, F. Fan, S. S. Mansouri, J. Abildskov and G. Sin, *J. Chem. Inf. Model.*, 2023, **63**, 725–744.
77 R. Li, J. M. Herreros, A. Tsolakis and W. Yang, *Fuel*, 2020, **280**, 118589.
78 Z. Liu, L. Shang, K. Huang, Z. Yue, A. Y. Han, D. Wang and H. Zhang, *Environ. Sci. Technol.*, 2025, **59**, 857–868.
79 X. Cao, M. Gong, A. Tula, X. Chen, R. Gani and V. Venkatasubramanian, *Engineering*, 2024, **39**, 61–73.
80 M. Jiang, G. Pedrielli and S. H. Ng, *Proceedings of the 2022 Winter Simulation Conference, WSC 2022*, 2022, 49–60.
81 E. A. Eugene, K. D. Jones, X. Gao, J. Wang and A. W. Dowling, *Comput. Chem. Eng.*, 2023, **179**, 108430.
82 D. T. Agi, K. D. Jones, M. J. Watson, H. G. Lynch, M. Dougher, X. Chen, M. N. Carlozo and A. W. Dowling, *Curr. Opin. Chem. Eng.*, 2024, **43**, 100994.
83 N. V. Sahinidis, M. Tawarmalani and M. Yu, *AIChE J.*, 2003, **49**, 1761–1775.
84 J. S. Rowlinson, *Liquids and Liquid Mixtures*, Butterworth, London, 1969.
85 J. M. Smith, H. C. Van Ness and M. M. Abbott, *Introduction to Chemical Engineering Thermodynamics*, McGraw-Hill Education, 8th edn, 2017.
86 Y. A. Cengel and M. A. Boles, *Thermodynamics: An Engineering Approach*, McGraw-Hill Education, 8th edn, 2015.
87 M. Li, H. Wang, Z. Yang, L. Zhang and Y. Zhu, *Comput. Struct. Biotechnol. J.*, 2023, **21**, 5544–5560.
88 C. Shi and T. B. Borchardt, *ACS Omega*, 2017, **2**, 8682–8688.
89 *PubChemPy Documentation — PubChemPy 1.0.4 Documentation*, https://pubchempy.readthedocs.io/en/latest/.
90 *RDKit: Open-source cheminformatics*, `https://www.rdkit.org`, Accessed: 2024.
91 National Institute of Standards and Technology, Office of Data, *NIST Chemistry WebBook*, https://webbook.nist.gov/chemistry/, 2024.
92 R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*, Chapman Hall/CRC, Boca Raton, Florida, 2020.
93 M. G. Genton, *J. Mach. Learn. Res.*, 2002, **2**, 299–312.
94 A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani and J. Hensman, *J. Mach. Learn. Res.*, 2017, **18**, 1–6.
95 P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, *Nat. Methods*, 2020, **17**, 261–272.
96 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
97 K. Sechidis, G. Tsoumakas and I. Vlahavas, *Mach. Learn. Knowl. Discov. Databases*, 2011, 145–158.
98 P. Szymański and T. Kajdanowicz, Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications, ECML-PKDD, Skopje, Macedonia, 2017, pp. 22–35.
99 D. O. Abranches, Y. Zhang, E. J. Maginn and Y. J. Colón, *Chem. Commun.*, 2022, **58**, 5630–5633.
100 B. E. Turner, C. L. Costello and P. C. Jurs, *J. Chem. Inf. Comput. Sci.*, 1998, **38**, 639–645.
101 Y. Que, S. Ren, Z. Hu and J. Ren, *Processes*, 2022, **10**, 577.
102 J. Ferraz-Caetano, F. Teixeira and M. N. D. S. Cordeiro, *Chemosphere*, 2024, **359**, 142257.
103 Y. Beghour and Y. Lahiouel, *Chem. Eng. Sci.*, 2025, **309**, 121228.
104 L. D. Hughes, D. S. Palmer, F. Nigsch and J. B. O. Mitchell, *J. Chem. Inf. Model.*, 2008, **48**, 220–232.
105 V. Venkatraman, S. Evjen, H. K. Knuutila, A. Fiksdahl and B. K. Alsberg, *J. Mol. Liq.*, 2018, **264**, 318–326.
106 A. A. Pérez Ponce, I. Salfate, G. Pulgar-Villarroel, L. Palma-Chilla and J. A. Lazzús, *J. Engin. Thermophys.*, 2013, **22**, 226–235.
107 H. Wiener, *J. Am. Chem. Soc.*, 1947, **69**, 17–20.
108 I. Gutman and N. Trinajstić, *Chem. Phys. Lett.*, 1972, **17**, 535–538.

109 M. Randic, *J. Am. Chem. Soc.*, 1975, **97**, 6609–6615.
110 E. A. Eugene, W. A. Phillip and A. W. Dowling, *Curr. Opin. Chem. Eng.*, 2019, **26**, 122–130.