Hybrid, Interpretable Machine Learning for Thermodynamic Property Estimation using Grammar2vec for Molecular Representation

Vipul Mann[a], Karoline Brito[b], Rafiqul Gani[c], Venkat Venkatasubramanian[a,*]

[a] *Department of Chemical Engineering, Columbia University, New York, USA*
[b] *Chemical Engineering Department, Federal University of Campina Grande, Brazil*
[c] *PSE for SPEED Company, Ordrup Jagtvej 42D, DK-2920, Charlottenlund, Denmark*

Abstract

Property prediction models have been developed for several decades with varying degrees of performance and complexity, from the group contribution-based methods to molecular simulations-based methods. An interesting issue in this area is finding an appropriate representation of molecules inherently suited for the property modeling problem. Here, we propose Grammar2vec, a SMILES grammar-based framework for generating dense, numeric molecular representations. Grammar2vec embeds molecular structural information contained in the grammar rules underlying SMILES string representations of molecules. We use Grammar2vec representations to build machine learning-based models for estimating normal boiling point (T_b) and critical temperature (T_c) and benchmark their performance against the popularly used group contribution (GC)-based methods. To ensure interpretability of the developed ML model, we perform a Shapley values-based analysis to estimate feature importance and simplify (or prune) the trained model.

1. Introduction

Thermodynamic properties are crucial for efficient process design, optimization, control, and monitoring, the latter two being extremely important in safety-critical conditions. A reliable (and often preferable) means of acquiring properties is measuring them through controlled experimentation that achieves the desired level of accuracy and precision. However, with increasing complexity and number of compounds synthesized at a rapid pace, along with a combinatorially large number of possible mixtures, it is nearly impractical to perform costly and time-consuming experiments for

*Corresponding author
Email address: venkat@columbia.edu (Venkat Venkatasubramanian)

all of them. An alternative solution for mitigating this issue is to build fairly accurate data-driven models that could be used to estimate such properties. As argued by Venkatasubramanian[1], the most useful data-driven methods are those that combine domain knowledge (in the form of symbolic information) with numeric machine learning.

Quantitative structure-activity/property relationships (QSAR/QSPR) methods utilize the correlations between molecular properties and their structural descriptors for data-driven property estimation. Although, these methods are statistical in nature and are characterized by difficult mathematical formulations and property-specific nature that limited their wider generalization capabilities [2], the commonly used Group Contribution (GC)-based methods, which could be regarded as a special class of QSAR/QSPR methods, are simple, easy to use and predictive in nature because the same contributions of a group can be used in any molecule they represent. The limitations of the GC-based methods, however, in terms of accuracy, applicability to complex molecular structures, isomer distinction, etc., are also well known [3]. The underlying assumption in group contribution methods is that the property of a compound is a function of its molecular structure and the property can be determined by summing the contributions of the groups representing the molecule for a specific property. Well known GC methods for pure component properties prediction include those of Lydersen [4], Joback and Reid [5], Constantinou and Gani [6], Benson et al. [7], Marrero and Gani [8], Hukkerikar et al. [9], to name a few. GC methods related to phase equilibrium computations include excess Gibbs energy based methods such as UNIFAC [10], DISQUAC [11] and MOSCED [12], and equations of state-based methods, such as predictive simple PC-SAFT [13]. A more complete list of GC methods for different properties can be found in review papers such as Katritzky et al. [14], Gmehling [15] and Gani [3].

Recently, machine learning (ML) methods have emerged as powerful alternative for tackling the thermodynamic property estimation problem. These methods generally use artificial neural networks, support vector regression, or deep learning approaches involving autoencoders, variational autoencoders, graph neural networks, and so on. Such methods have been reportedly used for predicting CO_2 solubility, density and viscosity of potassium lysinate and the mixed solutions with monoethanolamine [16], predicting standard enthalpies of formation for hydrocarbons [17], density and viscosity of biofuel compounds [18], and and, GC-based machine learning modeling of 25 pure component properties of organic compounds [19].

One of the important steps in building such data-driven ML models is using an appropriate representation for molecules that captures their underlying structural and chemical characteristics. This task could either be performed manually based on domain knowledge by using expert-curated features for each property individually, or performed automatically in a latent space, referred to as representation learning [20]. Such representations (or molecular features) are fed to the data-driven models with the target variable as the property of interest for a molecule. In the area of chemistry and drug discovery, the common representations for molecules are Morgan fingerprints [21], SMILES strings [22], molecular graphs [23], and SMILES grammar [24]. Recently, SMILES grammar-based representations [25, 26] were proposed for the forward reaction prediction and the retrosynthetic analysis problems. These representations were shown to incorporate rich molecular-structure information, overcome overfitting in ML models, and are superior from an information-theoretic standpoint [26]. A similar analysis on property estimation is presented in this paper (see Section 5.1). At the present, there is no consensus on which molecular representation is most suited for machine learning-based property prediction problems, making this an interesting field of research.

In this paper, we evaluate the use of the Grammar2vec framework, which generates dense vector representations of molecules using the grammar rules defining SMILES representations to develop ML-based property estimation models. We evaluate the performance of these models in terms of SMILES and SMILES grammar-based representation of molecules. We benchmark their performance on two thermodynamic properties – normal boiling point (T_b) and critical temperature (T_c). To ensure a fair comparison (fixing the number of features in each representation), we invoke the natural language analogy and look at molecules as sentences and the underlying atomic units as words to generate their fixed-size vector representations. These vector representations are the same for each molecule (irrespective of the property being predicted), which are then used to build separate ML-based regression models. In addition, we perform a systematic study using Shapley values [27] to understand the feature importance and prune the model further, giving rise to a simpler and relatively interpretable model. We refer to this approach of generating dense vector representations of molecules using the grammar rules underlying the SMILES representation as the Grammar2vec framework.

The rest of the paper is organized as follows – in Section 2, we formally define the property

estimation problem along with an algorithmic overview of the proposed approach; Section 3 provides an overview of the various methods including SMILES and SMILES grammar-based molecular representations in Section 3.1, the proposed Grammar2vec framework in Section 3.2,details on the modeling framework using kernel SVR in Section 3.3, and the Shapley values formulation for model interpretability in Section 3.4. A description of the dataset and model training aspects including hyperparameter tuning and feature visualization is provided in Section 4. The detailed results are presented in Section 5 with an information-theoretic analysis for comparing the representations in a machine learning-independent manner in Section 5.1, regression statistics for the property prediction task and comparison with GC-based methods in Section 5.2, and feature importance and model pruning obtained using Shapley values in Section 5.3 and 5.4, respectively. Finally, the conclusions from this study and future outlook for this problem appear in Section 6.

2. Problem statement and objectives

We formulate the thermodynamic property estimation problem as a regression task where the objective is to build a machine learning-based regression model between the regressors and the target variable as,

$$\hat{y}_i = \mathbf{f}(\mathbf{x_i}, \boldsymbol{\beta}) + e_i \tag{1}$$

where (for the i^{th} molecule) the target variable $\hat{y}_i$ is the predicted thermodynamic property of interest; $\mathbf{x_i}$ is the vector representation of the molecule and is of dimension $m \times 1$; $\boldsymbol{\beta}$ is a vector of regression coefficients (with appropriate dimensions depending on the model-form) that is estimated; and e_i is the noise in the prediction that could be attributed to the measurement errors in the training data and/or the modeling inaccuracies. The former could be inferred by performing an uncertainty analysis as demonstrated for GC-models in [9].

Based on Equation 1 above, the two important aspects for estimating the target value y_i accurately are – choosing an appropriate functional transformation $\mathbf{f}(.)$ and a using rich molecular representation $\mathbf{x_i}$. For modeling the former, we use a support vector regression (SVR) model with radial basis function (RBF) kernel and for the latter, generate a rich, property-agnostic molecular representation $\mathbf{x_i}$ using the proposed Grammar2vec framework. In addition, since our objective is to build interpretable machine learning models, we analyze the feature importance and contribu-

tion towards the predictions by computing Shapley values. This information is used to prune the model and simplify it as much as possible while retaining a similar performance, thus giving rise to a relatively simpler yet powerful model architecture. The kernel SVR model, the Grammar2vec framework, and Shapley values are described in detail in Sections 3.3, 3.2, and 3.4 respectively. An overview of the proposed algorithmic framework is presented in Figure 1.

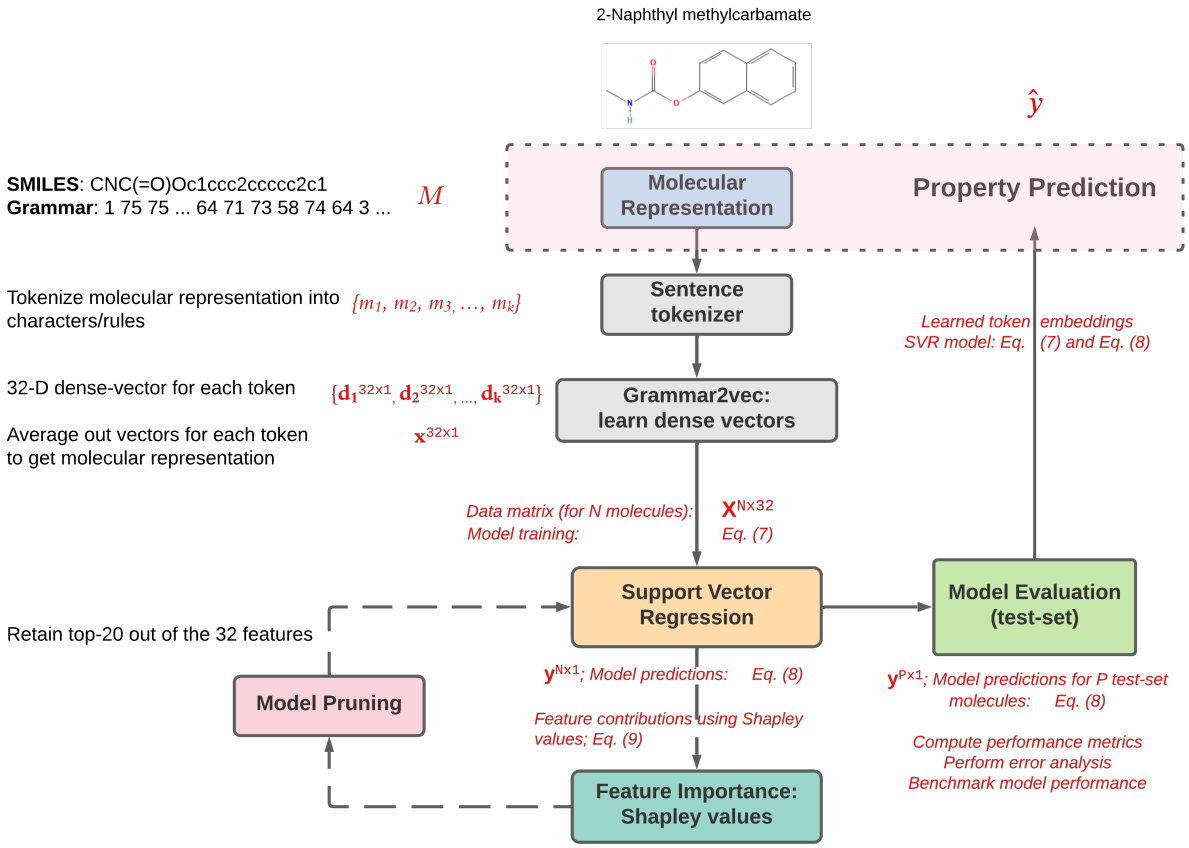


Figure 1: An overview of our algorithmic framework used for the property prediction problem. The details on various components are presented in Section 3.

For the example molecule 2-Napthyl methylcarbamate, the two different representations based on the SMILES and SMILES grammar representations are indicated in Figure 1. These representations are tokenized (split into individual characters) and passed to two separate Grammar2vec models as 'words' that learns their respective 32-dimensional (dense) vector representations which are then transformed to a single 32-dimensional dense vector representing the entire molecule by performing element-wise averaging. These 32 features are then used in the kernel SVR model to estimate the property of interest (say T_c). While this model could directly be used for property prediction, we

simplify the model by first computing the Shapley values to understand the relative importance of the 32 features towards the regression task and then prune the model to drop the relatively unimportant features. The final, simplified model is then used for the property prediction task. The equations and transformations required at each stage are indicated in Figure 1. The following sections provide details on each of these modules.

3. Methods underlying grammar2vec-based property estimation

In this section, we provide a brief overview of the various methods used in our work at different stages in the algorithmic framework described in Figure 1 above, including the different molecular representations, the natural language analogy and the proposed Grammar2vec framework, the kernel SVR model used for estimating the properties, and the Shapley values approach used for computing feature importance and performing model pruning.

3.1. Molecular representations: SMILES and SMILES grammar

The SMILES representation [22] is commonly used in deep-learning approaches including drug discovery, because of their ability to encode molecular structural information as text using short ASCII strings. Several approaches that have utilized this representations in frameworks adapted from natural language process include the SMILES transformer [28], Mol2vec [29] for generating molecular representations inspired from [30], and SMILES2vec [31] for chemical property estimation using deep recurrent neural networks, and several other works on property estimation [32–35]. The major shortcoming of the SMILES representation, however, is their lack of explicit incorporation of the entire molecular structure of the molecule (including the 3D structure and stereochemistry). It is often assumed (incorrectly) that the machine learning model would discover the underlying relationships between the SMILES characters, resultig in information loss and consequently suboptimal model performance. This issue is addressed to a large extent by utilizing the underlying SMILES grammar production rules.

The SMILES grammar specifies the underlying production rules that are required to generate the SMILES string of a given molecule. These productions rules are much more detailed and richer when compared to individual, purely character-based SMILES characters (or tokens). The SMILES grammar is similar to the context-free grammar (CFG) commonly used in the area of natural

Table 1: A subset of the SMILES grammar productions. The complete SMILES grammar used in this work is presented in the Appendix A.

S.No	Production rules
1	SMILES ⟶ CHAIN
2	CHAIN ⟶ CHAIN BRANCHED_ATOM
3	CHAIN ⟶ CHAIN BOND BRANCHED_ATOM
4	CHAIN ⟶ BRANCHED_ATOM
5	BRANCHED_ATOM ⟶ ATOM RINGBOND
6	BRANCHED_ATOM ⟶ ATOM
7	BRANCHED_ATOM ⟶ ATOM BB
8	BRANCHED_ATOM ⟶ ATOM RB
9	BB ⟶ BRANCH
10	RB ⟶ RINGBOND
11	BRANCH ⟶ (CHAIN)
12	RINGBOND ⟶ DIGIT
13	BOND ⟶ =
14	ATOM ⟶ AROMATIC_ORGANIC
15	ATOM ⟶ ALIPHATIC_ORGANIC
16	AROMATIC_ORGANIC ⟶ c
17	ALIPHATIC_ORGANIC ⟶ C
18	ALIPHATIC_ORGANIC ⟶ O

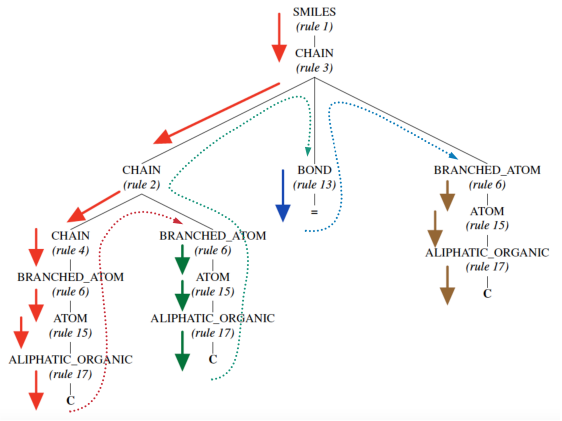


Figure 2: The parse-tree obtained for propene (SMILES representation: CC=C) using SMILES grammar productions in Table 1. The productions are extracted from the parse tree in a depth-first manner, resulting in the grammar representation for propene to be 1, 3, 2, 4, 6, 15, 17, 6, 15, 17, 13, 6, 15, 17 (figure from Mann and Venkatasubramanian [25]).

language processing, first formalized by Chomsky [36]. A CFG is a finite collection of recursive rules (or productions) that defines the set of all well-formed sentences in a language. Formally, a CFG is a 4-tuple $G = (V, \Sigma, R, S)$, where V is a finite non-empty set of non-terminal symbols, Σ is a finite set of terminal symbols, R is a finite non-empty set of rules, and S is a designated start symbol. Each rule has a left-hand side (a single non-terminal) and a right-hand side (a sequence of one or more non-terminal or terminal symbols). A CFG can also be deduced for SMILES strings and was used for molecule optimization in [24] in which the parse tree is used as a representation for the molecules [25]. A representative subset of the SMILES grammar is shown in Figure 1 and the set of rules required to generate the SMILES string for propene (CC=C) is shown in Figure 2 in the form of a parse tree. A SMILES grammar-based representation could be constructed from the grammar parse tree by extracting the sequence of production rules in a depth-first strategy as indicated in the caption of Figure 2. Recently, Mann and Venkatasubramanian [25, 26] showed that the SMILES grammar-based representations could be adapted for the forward reaction prediction involving multiple reactants, agents, and product molecules, and showed that this approach perform well even for single-step retrosynthesis prediction.

The SMILES grammar-based representations incorporate chemical and structural information about molecules, which cannot be done with purely character-based SMILES strings, and are shown to be superior from an information-theoretic standpoint [26]. The grammar rules, that often cor-

respond to underlying structural chemistry of molecules, explicit encode that information in the representation making them much more richer as compared to SMILES strings. The complete SMILES grammar used in our work is presented in the Appendix. We transform these grammar rules into dense, numeric vectors using the Grammar2vec framework as described in the next section.

3.2. Grammar2vec framework

Natural language processing techniques involve transforming text into its equivalent numeric representation that preserves the underlying properties (context, meaning, structure) to the maximum possible extent. One approach to achieve this is generating word embeddings that learn dense vector representations of text (or words) using neural networks by preserving the contextual information of words in a corpus. Word2vec [30] is one of the most commonly used methods used to generate word embeddings and is at the core of the Grammar2vec framework. It encodes words in a high dimensional vector space such that words with higher semantic similarity are closer in the vector space. The Word2vec model can be trained using two different approaches – skip-gram and a continuous bag of words (CBOW). The skip-gram approach involves using the current word to predict its contextual words, whereas the CBOW approach involves predicting the current word based on the context words. In our work, we work with the CBOW approach because of its better performance and faster computation.

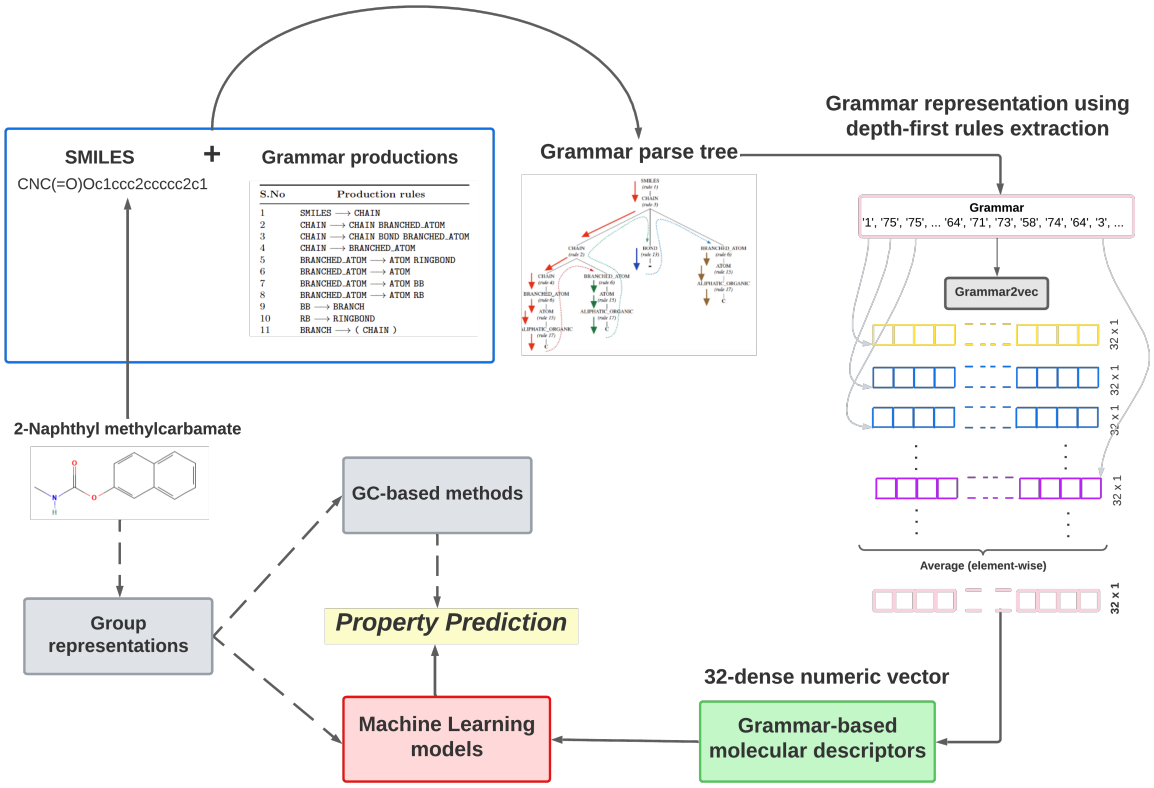


Figure 3: An overview of the proposed Grammar2vec framework for generating dense, numeric vector representation of molecules utilizing the underlying structural information in the form of SMILES grammar productions. The grammar-based descriptors are fed to machine learning models as features (or regressors). The other alternatives are using group representations with GC-based methods, or as features for ML models [19].

The steps involving the Grammar2vec framework are highlighted in Figure 3. Grammar2vec generates vector embeddings of molecules by considering grammar-representations of molecules as sentences and the individual grammar rules as separate words. For instance, consider the compound 2-Naphthyl methylcarbamate with SMILES string 'CNC(=O)Oc1ccc2ccccc2c1'. The 'molecular sentence' based on the grammar representation is generated by using the SMILES grammar representation as

'1' '75' '75' '75' ... '74' '64' '3' ... '3' '7' '66' '3' '6' '71' ... '39' '81'

where each production rule is equivalent to the 'molecular word'. Treating these representations as sentences and the individual constituent units as words, we train the Word2Vec model using the gensim package in Python [37] and learn the vector representations of each of these molecular words. The vector representation for the entire molecule (molecular sentence) is obtained by averaging out

the word embeddings learned across all the constituent words, a standard approach for learning sentence embeddings in natural language processing. A similar approach involving purely character-based SMILES strings (without grammar) is the idea behind SMILES2vec [31]. Similarly, [38] used Word2Vec model with bond-strings as input to generate molecular descriptors that were used for property estimation in a deep learning framework.

Since we compare the performance of Gramamr2vec representation against SMILES strings-based dense representation, we train two separate Word2vec models on a corpus of nearly $\sim 15,000$ molecules for $10,000$ iterations to learn dense vector representations of size 32 from molecular sentences.

Remark 1: Note that the actual characters in the molecular sentences do not matter; instead their relative position and the context words are important while learning their word embeddings. Therefore, we could replace the individual tokens to any arbitrary string as long as they map to the same word uniquely.

Remark 2: Just like the group contribution (GC)-based approaches, if a molecule is not present in the dataset, we can still generate its vector representation since the dense vectors are obtained using models trained on the individual bits in the molecular sentences comprising SMILES grammar rules or the SMILES characters. Therefore, any new molecule could be represented as long as these individual bits that it is composed of are seen by the model during training (a fair assumption of fixed vocabulary size, as in natural language). In fact, in our test set, we have molecules that were never seen by the model during the training stage.

3.3. Support vector regression for property estimation

To estimate the thermodynamic properties of molecules, we use a kernel support vector regression (SVR) model [39] to build a regression model between the molecular descriptors and the correspond property values. We chose the kernel SVR framework due to its ability to model complex, non-linear interactions between features even under small sample conditions and generalizability to out-of-sample datasets. The general idea behind the kernel SVR method is to first map the original data into a high dimensional space using kernels, and then find an optimal decision boundary or separating hyperplane by minimizing the error or the distance between the observed and the predicted values by formulating this as a constrained optimization problem.

For the simple linear SVR case (we use non linear SVR described in subsequent paragraphs), the regression function $\mathbf{f}(.)$ in Equation 1 is assumed to be linear and is defined as,

$$\mathbf{f}(\mathbf{x_i}, \boldsymbol{\beta}) = \mathbf{x_i}^T \boldsymbol{\beta} + b \tag{2}$$

where b is an additional coefficient that corresponds to the bias in the predictions, $\mathbf{x_i}$ is a numeric vector representing a molecule, and $\boldsymbol{\beta}$ is a vector of unknowns that need to be estimated. The objective is to minimize the norm of the coefficients $\boldsymbol{\beta}$ such that the residuals are within a given limit ϵ. Slack variables are introduced for each point in the constraints equation to ensure the constraints are satisfied and a solution exists for the constrained optimization problem (known as soft-margin SVR). The primal formulation of the optimization problem is thus given by,

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{\beta}||^2 + C \sum_{i}^{N} (\zeta_i + \zeta_i^*) \tag{3}$$

subject to,

$$y_i - (\mathbf{x_i}^T \boldsymbol{\beta} + b) \leq \epsilon + \zeta_i, \quad (\mathbf{x_i}^T \boldsymbol{\beta} + b) - y_i \leq \epsilon + \zeta_i^*$$

where, C is a parameter that controls the penalty imposed on points that lie outside the ϵ margin, and ζ_i and ζ_i^* are non-negative slack variables that define the maximum tolerable error without imposing a penalty on the regression errors. The Lagrange dual formulation for this problem, that is computationally easier to solve, is given by,

$$\min_{\lambda} \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (\lambda_j - \lambda_j^*)(\lambda_k - \lambda_k^*)(\mathbf{x_j}^T \mathbf{x_k}) + \epsilon(\lambda_j + \lambda_j^*) - \sum_{j=1}^{N} y_j(\lambda_j - \lambda_j^*) \tag{4}$$

subject to,

$$\sum_{j=1}^{N} (\lambda_j - \lambda_j^*) = 0, \quad 0 \leq \lambda_j \leq C, \quad 0 \leq \lambda_j^* \leq C$$

where λ_j and λ_k are non-negative Lagrange multipliers. The regressed (predicted) values, $\hat{y}_i$, for an

input $\mathbf{x_i}$, is given by

$$\hat{y}_i = \sum_{j=1}^{N} (\lambda_j - \lambda_j^*)(\mathbf{x_j}^T \mathbf{x_i}) + b \tag{5}$$

Instead of using the linear assumption for the regression function $\mathbf{f}(.)$ in Equation 1, we use a kernel function that transforms the data into higher dimensions to learn a non-linear function in the original space. This is done by replacing the inner products or dot products in the linear SVR formulation above with kernels. We use the radial basis function (RBF) kernel which is given by,

$$K(x_j, x_k) = exp(-\gamma ||x_j - x_k||^2) \tag{6}$$

where the parameter γ controls the width of the kernel. The Kernel matrix is an $n \times n$ matrix where each element corresponds to the inner product of the transformed data points in higher dimensions. Replacing the inner products with the RBF kernel function, the dual problem thus becomes,

$$\min_{\lambda} \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (\lambda_j - \lambda_j^*)(\lambda_k - \lambda_k^*) K(x_j, x_k) + \epsilon(\lambda_j + \lambda_j^*) - \sum_{j=1}^{N} y_j(\lambda_j - \lambda_j^*) \tag{7}$$

subject to,

$$\sum_{i=1}^{N} (\lambda_j - \lambda_j^*) = 0, \quad 0 \leq \lambda_j \leq C, \quad 0 \leq \lambda_j^* \leq C$$

The regressed (predicted) values, $\hat{y}_i$, for an input $\mathbf{x_i}$, is given by

$$\hat{y}_i = \sum_{j=1}^{N} (\lambda_j - \lambda_j^*)\ exp(-\gamma ||\mathbf{x}_j - \mathbf{x}_i||^2) + b \tag{8}$$

Therefore, Equations 7 and 8 characterize our trained non-parametric regression model, and the regressed values for thermodynamic properties of a give molecule could be estimated using Equation 8 for a given molecule represented as $\mathbf{x_i}$.

3.4. Interpretable ML using Shapley values

To improve the interpretability of the kernel SVR model that we use in the regression task, we are interested in understanding the feature importances and their contribution towards the estimated values. Such analysis is necessitated due to the inherent complexity of the kernel SVR model, primarily due to the high-dimensional feature transformation using the radial basis function (RBF) kernel. This renders the straightforward evaluation of the support vectors for understanding the feature importance nearly impossible. We, therefore, use Shapley values, a concept from cooperative game theory used to compute the contribution of each player to the final payout, to understand the feature importances for the property estimation task [40].

Shapley values are a measure of the average marginal contribution of a feature across all possible coalitions (or feature combinations). To quantify the importance of a given feature, different feature coalitions are simulated and the predicted value for the different contributions are averaged and subtracted from the predicted value with the given feature in the coalition. This computation is performed for all possible coalitions, and the Shapley value is the average of all the marginal contributions to all possible coalitions. Formally, the Shapley value for a feature j is defined as,

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} (\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m)) \tag{9}$$

where $\hat{f}(x_{+j}^m)$ is the prediction for x with a random number of feature values replaced by feature values from a random data point z except for the respective value of feature j; x_{+j}^m is identical to x_{-j}^m except that the value x_j^m is taken from the random sample z in x_{-j}^m; features on the left of x_j have values from the original observations and those on the right of x_j take their values from a random instance; and M is the number of instances generated. This procedure is repeated M times for all the features and feature importances are computed.

These concepts have been applied in various areas including machine learning for understanding ML models [41–44]. Following a similar approach, we compute the Shapley values for our model and also use the inferences to prune the model by retaining only the important features contributing the most to the model performance as depicted in the algorithmic framework in Figure 1. The Shapley values were computed using the SHAP package in Python [40] and the results on this are presented in Section 5.3.

4. Dataset and model training

In this section, we provide a detailed overview of the dataset and the model training aspects. Specifically, we first provide a description of the dataset for the two properties of interest, the model training, and hyperparameter tuning aspects for the kernel SVR models, and a visual comparison of the learned feature vectors for the dense vector embeddings of the molecular words obtained using Grammar2vec.

4.1. Dataset description

The dataset that we used in our work is based on data used in the recent work by Alshehri et al. [19] for predicting normal boiling point (T_b) and critical temperature (T_c). The dataset contains experimentally measured values of these properties with molecules represented as SMILES strings along with their groups-based representations indicating the various first-order, second-order, and third-order functional groups and the number of times they are present in each molecule. There are 200 first-order groups, 150 second-order groups, and 74 third-order groups, and hence, a total of 424 different functional groups. The data on normal boiling point for molecules consists of 3510 different pure compounds (molecules) whereas the data for critical temperature is much smaller with just 858 molecules. These molecules were further preprocessed to remove molecules that either could not be processed by RDKit or were not parsed by the SMILES grammar. The resulting final data had 3488 and 800 molecules for T_b and T_c, respectively.

Though the framework that we present in our work is applicable for predicting any thermodynamic property of a molecule as long as there is enough training data available, we build models only for predicting T_b and T_c because these two properties provide enough variety of challenges that are generally encountered – modeling difficulty, limited data availability, and performance on out-of-sample examples. Moreover, limiting ourselves to just two (but important) properties would help in performing a detailed analysis of the underlying models and the molecular representations trained to make them more interpretable.

4.2. Gramamr2vec and regression model hyperparameters

As explained in Section 3.2, the Grammar2vec model was used to learn dense vector representations for molecules using SMILES grammar and Word2vec using SMILES strings. We trained

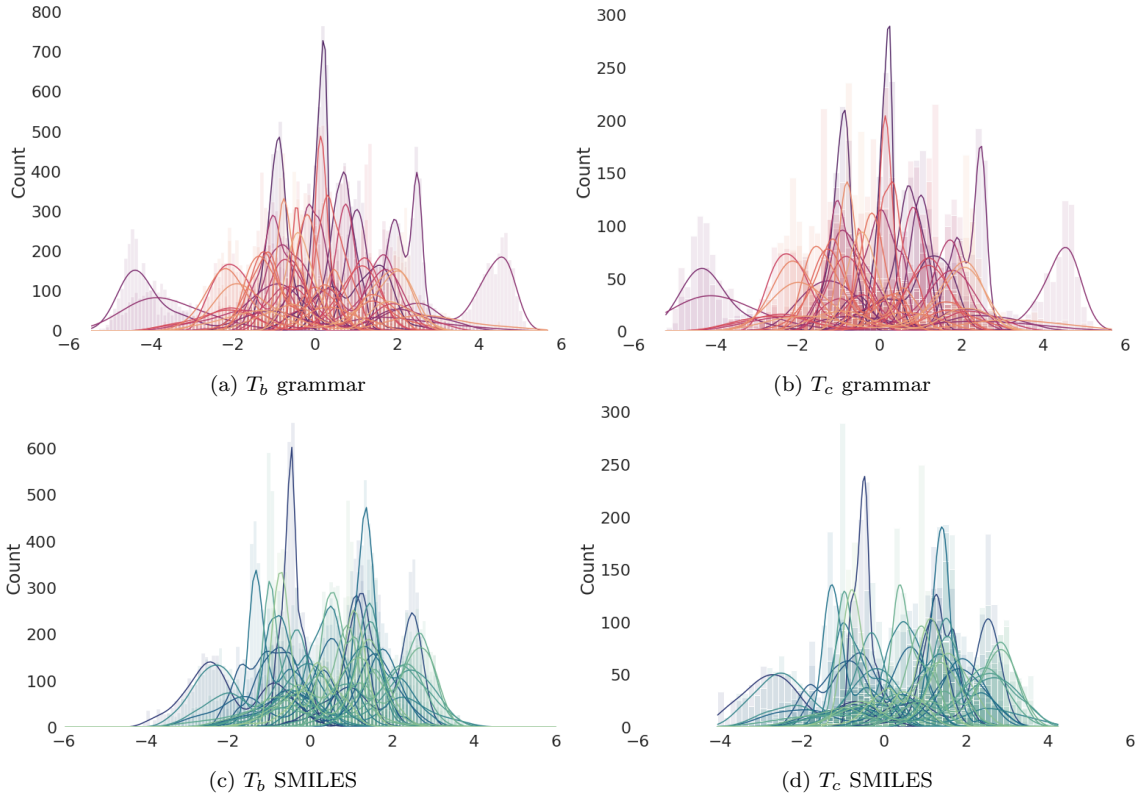
vector embeddings of molecules with different sizes, namely $8, 16, 32$, and 64 (resulting in the molecular representations $\mathbf{x}^{8\times1}, \mathbf{x}^{16\times1}, \mathbf{x}^{32\times1}$, and $\mathbf{x}^{64\times1}$, respectively), and evaluated the performance of learned representation of various sizes on a validation set in the property prediction task. The best model performance was achieved using an embedding of size 32, and hence, we fixed the size of the vector representations at 32 for performing final model training and detailed model analysis. These 32-dimensional dense feature vectors $\mathbf{x}^{32\times1}$ were used in the support vector regression (SVR) framework for training separate regression models for estimating T_b and T_c.

To perform modeling, we split our dataset into training and test sets using a 95/5 split where the training set is used for model building and the test set is used only for reporting model performance statistics. The training set is further subdivided to perform 5-fold cross-validation to search for optimal hyperparameter values using a randomized search in the following range: $C : 500-50000, \gamma : 0.01 - 0.2$, and $\epsilon : 0.1 - 1$. The hyperparameters are tuned separately for all the 4 models (using models trained on two separate representations each for T_b and T_c estimation). Therefore, each property prediction model is characterized by three hyperparameters that need to be tuned using the given dataset.

Remark 3: Since Grammar2vec is an unsupervised learning technique, we trained the word embedding models on the entire dataset containing nearly $\sim$ 15k molecules. One may choose to train this model on an even bigger dataset with millions of molecules such as the Zinc dataset or the USPTO reactions dataset.

4.3. Learned molecular representations

Since we are using a machine learning model for the regression task and using a 32-dimensional dense vector representation for molecules as input to the regression model, each individual feature should capture different aspects of a molecule. In other words, each feature should ideally focus on distinct aspects of a molecule, and that should be unique to a given feature. While there must be some overlap between what each feature captures, an ideal, rich representation would minimize this overlap (or similarities) across features. We qualitatively measure the similarity or dissimilarity across features by looking at their distribution plots (histograms). Higher the difference between the feature distributions, higher would be the richness of the representation. A rich representation, we hypothesize, would be the one that has the most condensed information in each feature.



(a) T_b grammar (b) T_c grammar

(c) T_b SMILES (d) T_c SMILES

Figure 4: Histograms for the 32 features based on the SMILES grammar and SMILES representations for molecules in the T_b and T_c datasets. The vertical axis corresponds to the frequency and the horizontal axis represents the actual numeric values of the feature vector elements.

The histograms for the dense vector-based representations for molecules learned using the Grammar2vec and Word2vec model (for SMILES) are shown in Figure 4. We plot the histograms for 32-dimensional molecular representations obtained using SMILES grammar and SMILES. There are 32 shades of colors in each sub-figure and each of them corresponds to a different feature.

5. Results

We present detailed results obtained using our algorithmic framework in this section. We first present an information-theoretic analysis in Section 5.1 for comparing the various learned representations in an ML-independent manner that is rooted in fundamental analysis of uncertainty and conditional information gain associated with the representations. The detailed regression statistics, error analysis, and benchmarking the performance againts group contributions (GC)-based methods are presented in Section 5.2. The feature importance and subsequent model pruning results and analysis based on Shapley values are in Section 5.3 and Section 5.4, respectively.

5.1. Information theory

In representation learning, it is important to preserve as much information about the underlying entity as possible, and typically, the representations that are richer and have higher information perform well when used in machine learning frameworks [26]. Representations are important because they guide ML models in discovering the underlying, hidden patterns much more easily if a richer representation is used, as opposed to an obscure, difficult to interpret representation. We, therefore, turn to information theory since it offers a quantitative approach for quantifying the amount of information associated with a given communication channel [45] that could be adapted for chemical representations [26].

Two central concepts in information theory are the Shannon entropy and the conditional entropy (or information gain). Shannon entropy measures the amount of uncertainty associated with a given signal in terms of bits of information and by definition, a higher uncertainty translates to higher Shannon entropy or information capacity. Higher Shannon entropy points towards the higher information-carrying capacity of a communication channel. On the other hand, conditional information gain or conditional entropy is a measure of the amount of uncertainty associated with a signal when a part of it is known (hence, conditional). Typically, a lower conditional information gain is desirable since it translates to a lower reconstruction error or loss. Figure 5 depicts these two concepts schematically along with the mathematical equations for their computation.

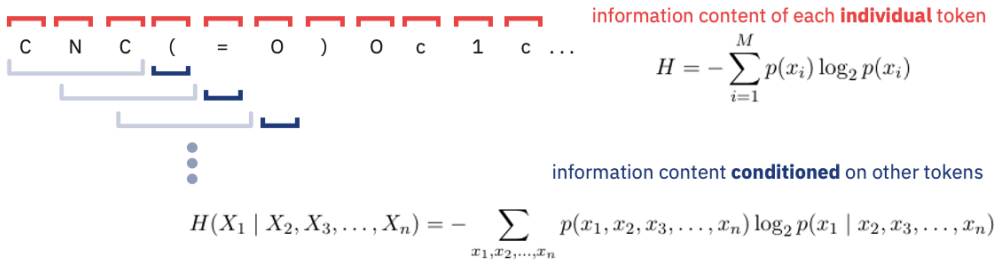


Figure 5: Shannon entropy and conditional entropy depicted schematically on the SMILES string representation for 2-Naphthyl methylcarbamate.

To utilize these concepts from information theory and analyze chemical representations from this perspective, we look at the individual tokens (or characters) in various representations as random variables, and therefore, the molecular representation becomes a sequence of random variables, $X_1, X_2, \ldots, X_n$, where n is the length of the representation for a given molecule and X_i could take any of the M possible tokens defined in the vocabulary of the representation. Tokens are the

individual production rules for the SMILES grammar representation, or characters for the SMILES representation. For instance, consider the same compound 2-Naphthyl methylcarbamate. The different representations for this compound along with the associated random variables or tokens are as follows:

- SMILES (‘C’ ‘N’ ‘C’ ‘(’ ‘ = ’ ‘O’ ‘)’ ‘O’ ‘c’ ‘1’ ‘c’ … ‘c’ ‘2’ ‘c’ ‘1’): $X_i^{SMILES} \in \{‘C’, ‘N’, ‘(’, ‘ = ’, ‘O’, ‘)’, ‘c’, ‘1’, ‘2’\}$, where $M = 9, n = 22$
- Grammar (‘1’ ‘75’ ‘75’ ‘75’ … ‘3’ ‘7’ ‘66’ ‘3’ ‘6’ ‘71’ … ‘39’ ‘81’): $X_i^{Grammar} \in \{‘1’, ‘3’, ‘4’, ‘6’, ‘7’, ‘8’, ‘16’, ‘39’, ‘40’, ‘58’, ‘62’, ‘64’, ‘65’, ‘66’, ‘69’, ‘71’, ‘73’, ‘74’, ‘75’, ‘81’\}$, where $M = 20, n = 77$

We compute the Shannon entropy and conditional information gain using the entire set of nearly 15k molecules in the dataset since this is an unsupervised approach and does not require the target property values. We estimate the required (conditional) probability distributions for the two representations (SMILES, and grammar) based on the co-occurrence matrices at different orders of conditioning (up to an order $\eta = 5$). An order $\eta = 1$ corresponds to the Shannon entropy, order $\eta = 0$ corresponds to Shannon entropy when the random variables follow a uniform distribution (theoretical limit of information capacity), and orders $\eta > 1$ correspond to conditional entropy when $\eta - 1$ preceding random variables are known and is computed using the conditional entropy formula in Figure 5.

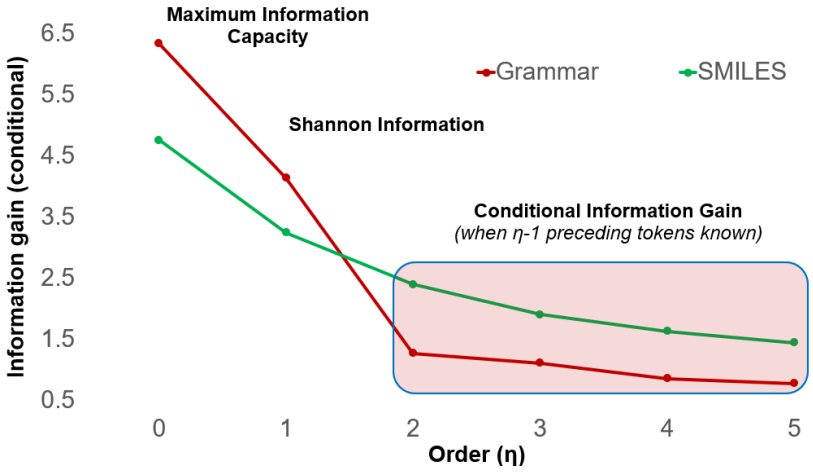


Figure 6: Information theoretic analysis results for the SMILES and grammar representations.

These results are presented in Figure 6 and we observe that even though the theoretical and

observed Shannon entropy is higher for the grammar representation, the conditional information gain drop significantly for the grammar representations for orders $\eta \geq 2$ and remains much lower than that of the SMILES representations even at increasingly higher orders. This points towards an important property of the grammar representations – when prior information (about preceding tokens) is known for the grammar representations, the amount of uncertainty associated with the other tokens for a molecular representation is reduced significantly which is not so much the case with SMILES representations. This could be attributed to the hierarchical structure (as shown in Figure 2) and the underlying grammar that is much more efficient in eliminating the infeasible tokens than the other representations.

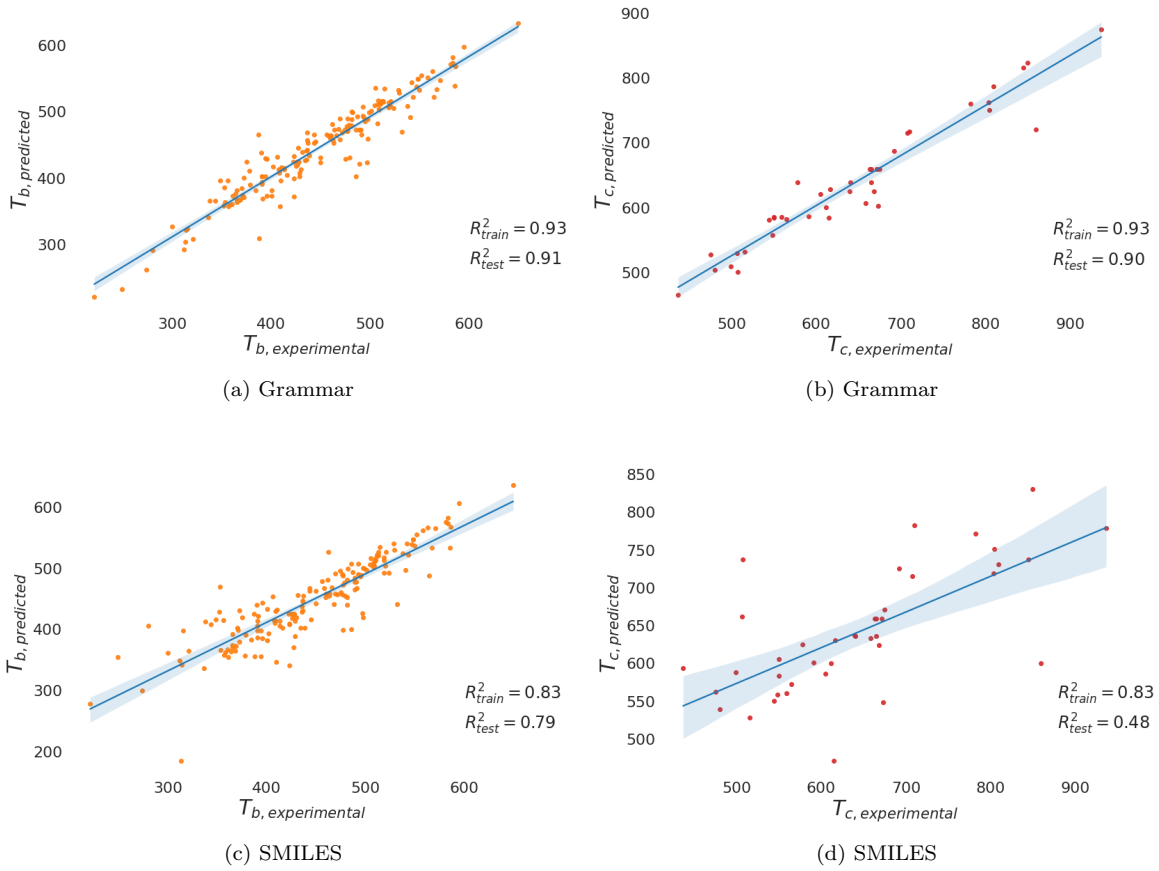
This property is extremely useful for methods like Word2vec (and consequently for Grammar2vec) because they attempt to encode a given word based on the contextual/neighboring words, and owing to the lower conditional information gain in the grammar representation, the contextual words are much more easier to infer. This property results in a lower reconstruction loss in the Word2vec model at the training stage and therefore, better representations. This richness and superiority of the grammar-based features should translate to better performance on the regression task.

5.2. Property model development and application

Now that we have compared the representations and established their relative richness using an ML-independent manner, we present the results on the regression task based on the kernel-SVR models (using equation 7) for each of the representations. Based on a random 95/5 split, the T_b dataset had 3313 data points in the training set and 175 data points in the test set. On the other hand, the T_c dataset had 760 data points in the training set and 40 data points in the test set. The scatter plots between the true values and the experimental values for normal boiling points (T_b) for molecules in the test set (unseen at the training stage) are presented in Figure 7. The R^2 values indicating the goodness of the predictions on the training and the test set are indicated in the inset of each sub-figure.

We observe that the grammar representations-based models have the highest R^2 values on the test set (comprising molecules unseen during the training stage), pointing towards their better generalization capabilities. In addition, their predictive accuracy is higher even when the number of

training samples is reduced by nearly an order of 4 (3488 for T_b vs 800 for T_c). The SMILES representations-based model is seen to perform poorly under small-sample conditions. The groups representations-based models though have a near-perfect accuracy on the training set, the performance on the test set in both the cases is significantly lower, and therefore, the models overfit the training set.



(a) Grammar (b) Grammar

(c) SMILES (d) SMILES

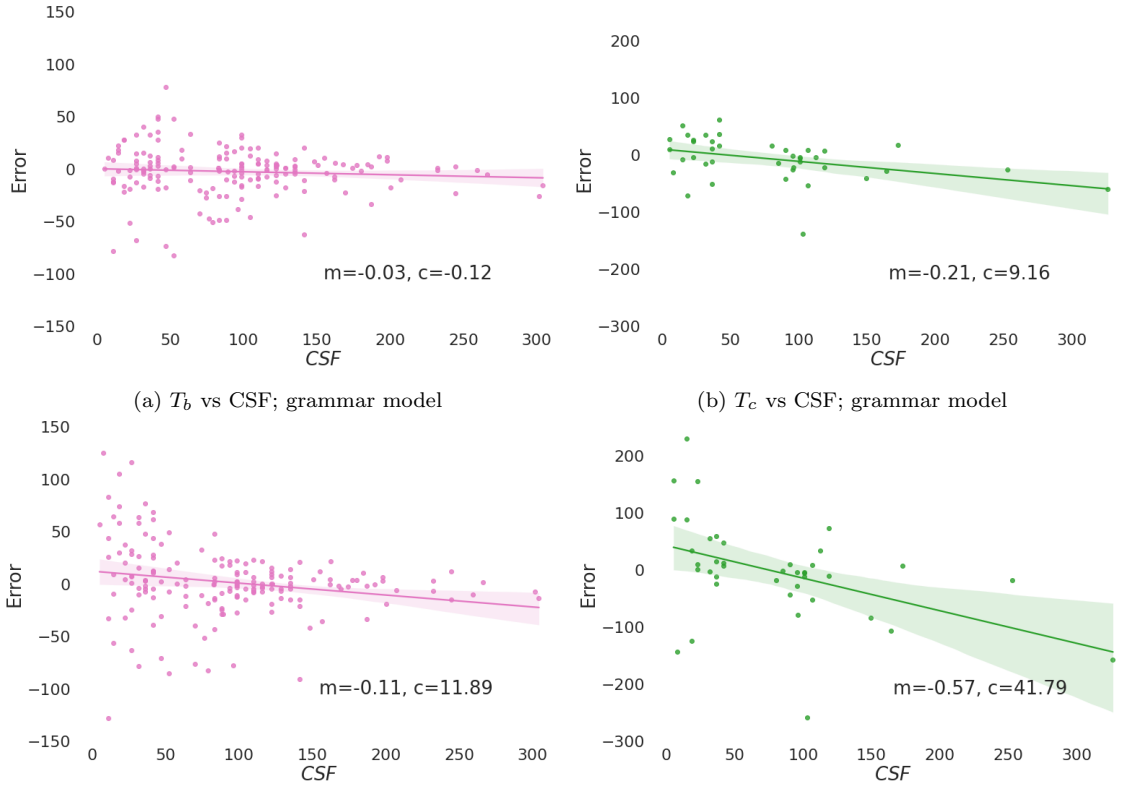
Figure 7: Regression results on the normal boiling point (T_b) and critical temperature (T_c) prediction tasks on the test set containing molecules unseen at the model training stage.

To gain a better understanding of the model performance for various molecules, we study the prediction errors ($\hat{y} - y$) as a function of the complexity of the molecule. We define the complexity of molecules inspired by the chemical scoring functions (CSF) defined in [46] for scoring the chemical and synthetic complexity of a reaction pathways as,

$$CSF = \text{SMILES\_LEN}^{3/2} + \alpha\text{RINGS} + \beta\text{STEREO} \tag{10}$$

where SMILES_LEN is the length of the SMILES string of the molecule which is related to its mass

and overall complexity, RINGS is the total number of rings in the molecule and STEREO is the number of stereocenters in the molecule. The parameters α and β could be tuned but for simplicity, we fix their values to the average value for the SMILES_LEN$^{3/2}$ across all molecules to ensure the three terms contribute roughly equally to the CSF score. The plot of the CSF and the prediction errors for T_b and T_c using different representations for molecules in the test set are presented in Figure 8.



(a) T_b vs CSF; grammar model

(b) T_c vs CSF; grammar model

(c) T_b vs CSF; SMILES model

(d) T_c vs CSF; SMILES model

Figure 8: Prediction error vs molecular complexity analysis using chemical scoring functions (CSF) for T_b and T_c on the test set containing molecules unseen at the model training stage.

Based on these results, we draw the following inferences:

- For the grammar representations, both for T_b and T_c, the relationship between the prediction error and CSF is either non-existent or very weak, pointing towards the complexity-agnostic nature and hence, wider generalizability/applicability of the fitted model. This is established by the slope and the intercept of the regression line along with their confidence bounds indicated in Figure 8(a) and (b) for T_b and T_c, respectively.

- The models fitted using the SMILES representations seem to underpredict the properties (negative $\hat{y} - y$) for complex molecules. This points towards their inability to capture the complex interactions between various functional groups in complex molecules and erroneously treats them as relatively simpler molecules. The grammar-based models do not suffer from this bias, possibly due to their richer features.

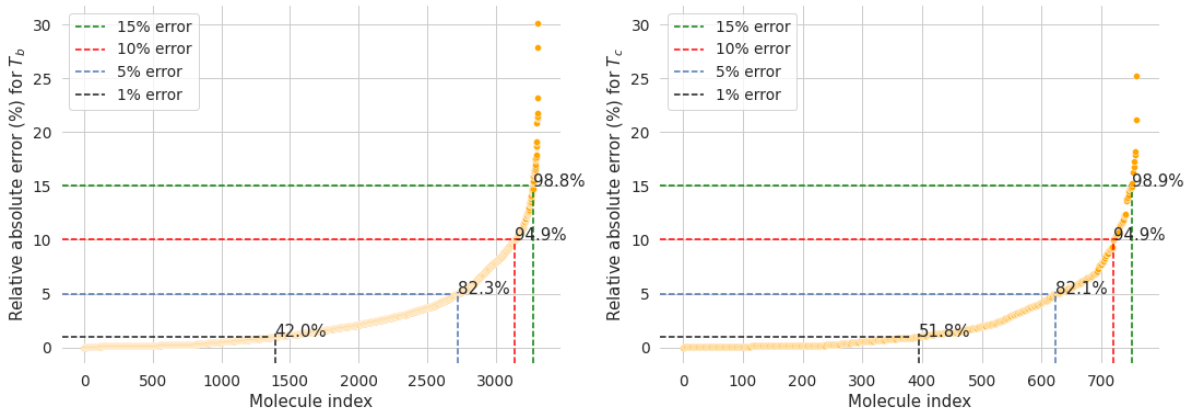
- Across both the models, the absolute values of the prediction errors are relatively high for smaller or simpler molecules. However, even in this case, the errors for the grammar-based models are much smaller than the errors in the other model.

We hypothesize that the nearly non-existent correlation between the prediction errors and the molecular complexity for the SMILES grammar model could be attributed to the higher structural information in-built in these representations. The sources of error in a machine learning models could be attributed to two factors – first, the structure-based errors and model inadequacies, and second, the errors in the parameter estimates. The latter is often due to limitations of the model in mapping the transformations between the features and the target variable including sub-optimal parameter estimates. The former, however, depends on the ability of the representation (features) in capturing the molecular chemistry. The grammar representations, therefore, do not seem to suffer from the structure-based limitations but only the model-based errors.

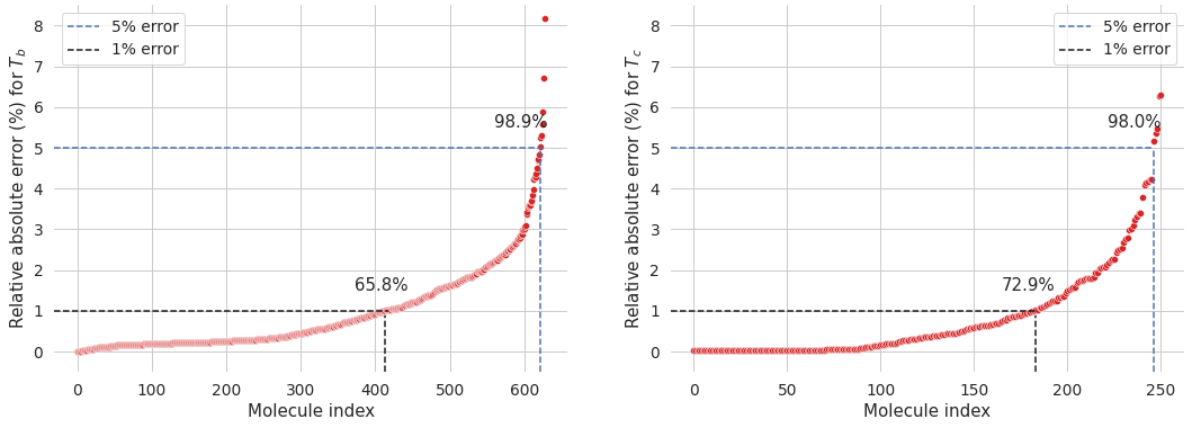
In order to asses the performance of our model better, we compute the percentage deviation of the predicted values from the measured property values, we compute the relative absolute percentage errors, for both T_b and T_c, defined as

$$RAE\ (\%) = 100 \times (y_{pred} - y_{true})/y_{true} \tag{11}$$

In addition, we also perform the same exercise end-to-end (representation learning, model training, RAE computation) for a subset of the entire dataset containing only hydrocarbon molecules. This dataset contained 1134 molecules that were used for learning molecular representations, 628 and 34 data points respectively for training and testing the $T_{b,hydrocarbons}$ estimation model, and 251 and 14 data points respectively for training and testing the $T_{c,hydrocarbons}$ estimation model. We present the error distribution plots for our models in Figure 9 and compare the results against those presented in Alshehri et al. [19] on various metrics in Table 2.



(a) Error distribution for the entire dataset



(b) Error distribution only for the hydrocarbons dataset

Figure 9: The relative absolute percentage error plots obtained using the grammar representation-based model for T_b and T_c on the entire dataset (training set + test set).

Table 2: Comparison with a previous work on property estimation that worked with the same dataset. The numbers indicate the percentage of molecules below the given percentage relative error threshold except the last column that provides values for the maximum percentage relative error observed in the predictions.

		1% error	5% error	10% error	15% error	Max. error
Alshehri et al. [19]	T_b	**79.1**	**87.8**	92.7	-	>50%
	T_c	**84.4**	**91.4**	94.2	-	>40%
Our model	T_b	42.0	82.3	**94.9**	**98.8**	**30.2%**
	T_c	51.8	82.1	**94.9**	**98.9**	**25.3%**
	$T_{b,hydrocarbons}$	65.8	**98.9**	**100**	**100**	**8.2%**
	$T_{c,hydrocarbons}$	72.9	**98.0**	**100**	**100**	**6.3%**

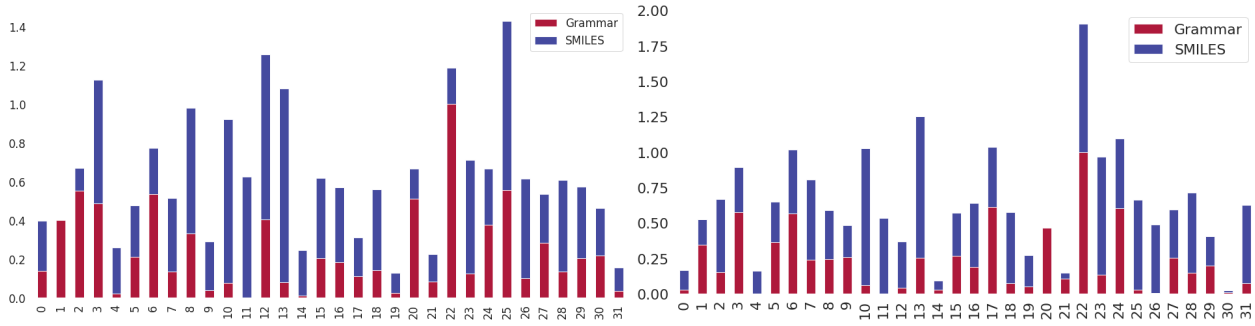
We observe that even though our model predicts relatively fewer number of molecules within the 1% error threshold, the model quickly catches-up and outperforms the model by Alshehri et al. for an error threshold of 10% and above, with $\sim$ 99% of all the predictions falling within a 15% error threshold. Moreover, the maximum prediction error for T_b and T_c at $\sim$ 30% and $\sim$ 25%, respectively,

is much lower than the maximum error reported by Alshehri et al. for their best performing model. The performance on the hydrocarbons dataset is significantly better with nearly $98-99\%$ of all the molecules within an error-threshold of less than 5%, both for T_b and T_c estimation. The gradually increasing trend in the relative errors (as opposed to a steep increase beyond a certain number of molecules), highlights a smooth functional mapping from the features to the property values learned by the ML-based regression model. As a consequence of this, the model has better generalizability. Therefore, we infer that our model is robust towards encountering new molecules or the presence of outliers in the dataset. The latter is a major issue because new molecules are being synthesized at a much faster pace and the experimentally measured properties dataset often have errors and outliers.

5.3. Feature importance and model interpretability

To further bolster the interpretability of our model, we perform an analysis using Shapley values to understand the contribution of the individual features towards the model predictions. The Shapley values were computed using a weighted kmeans approach using 10 samples weighted by the number of points they represent. We perform the analysis for the two models each for T_b and T_c properties, separately.

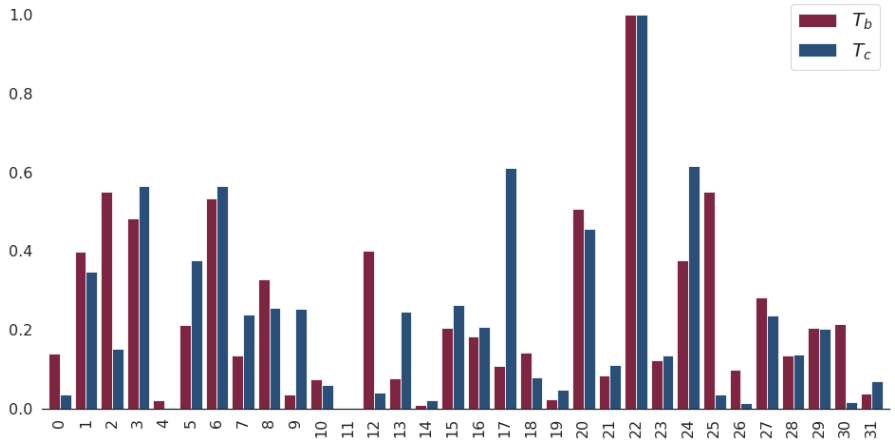
The feature contribution charts for the T_b and T_c predictions are presented as stacked bar plots in Figures 10 (a) and (b), respectively. In order to allow for an easy comparison on a relative scale, we have scaled the feature contributions in the range 0 to 1 by performing a min-max scaling for each model separately. We observe that the most important features (out of the 32 features) for the grammar model are T_b : $\{22, 25, 2, 6, 20, 3, 1, 12, 24, 8, 27, 5, 30, 15, 29, 16, 18, 28, 0, 7\}, T_c$: $\{22, 24, 17, 3, 6, 20, 5, 1, 15, 9, 7, 27, 13, 8, 29, 16, 2, 28, 23, 21\}$ and for the SMILES models are T_b : $\{13, 25, 12, 10, 3, 11, 8, 23, 26, 28, 18, 15, 7, 16, 29, 24, 0, 5, 27, 30\}$, T_c : $\{13, 10, 22, 23, 28, 25, 31, 24, 7, 2, 26, 11, 18, 17, 16, 6, 27, 3, 12, 15\}$.



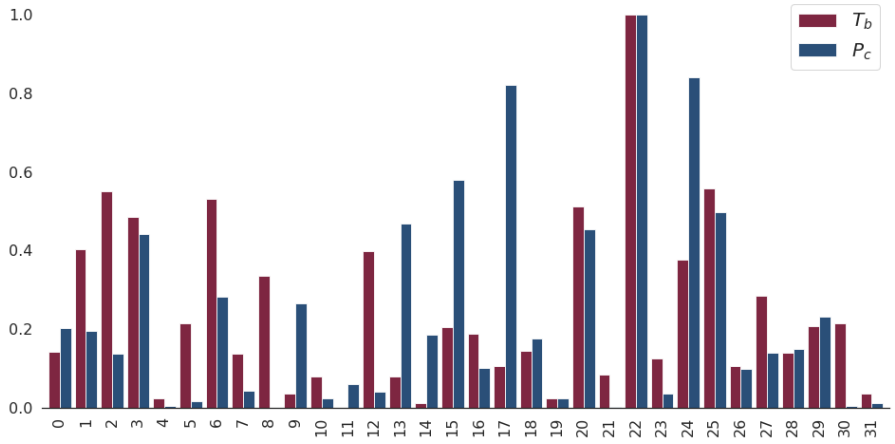
(a) T_b (b) T_c

Figure 10: Feature importance charts (scaled) for the T_b and T_c predictions for all the three representations. The feature importance are relative and indicate the contribution of each feature towards predicting the properties of interest. We see that not all features are equally important and the unimportant ones could be dropped without much impact on the model performance.

An interesting observation based on the above model is that the feature importance for a given molecular representation (grammar, SMILES) is very similar for both T_b and T_c predictions. For instance, of the 20 most important features for the grammar-based representations for T_b and T_c prediction, 15 out of 20 features are common. For the SMILES-based representations too 15 out of 20 features are common. This observation is rooted in chemistry since it is known that the normal boiling point (T_b) and critical temperature (T_c) of a given molecule are correlated. A side-by-side plot of the relative feature contributions for T_b and T_c models for grammar-based representations is shown in Figure 11 (a). In order to ensure this chemistry-correlation is indeed captured by the developed model and it is not just an artifact of the example considered, we train an additional grammar representation-based model following the same approach for predicting critical pressure (P_c) which has nearly the same number of training data points a T_c. A side-by-side plot of the relative feature contributions for T_b and P_c is shown in Figure 11 (b). As expected from chemistry, we see that the correlation between feature importance is much weaker between these properties when compared to the feature correlation between T_b adn T_c. Thus, we conclude that the grammar-based features preserve the underlying chemistry correlations between molecules that consequently in the model analysis.



(a) T_b vs T_c

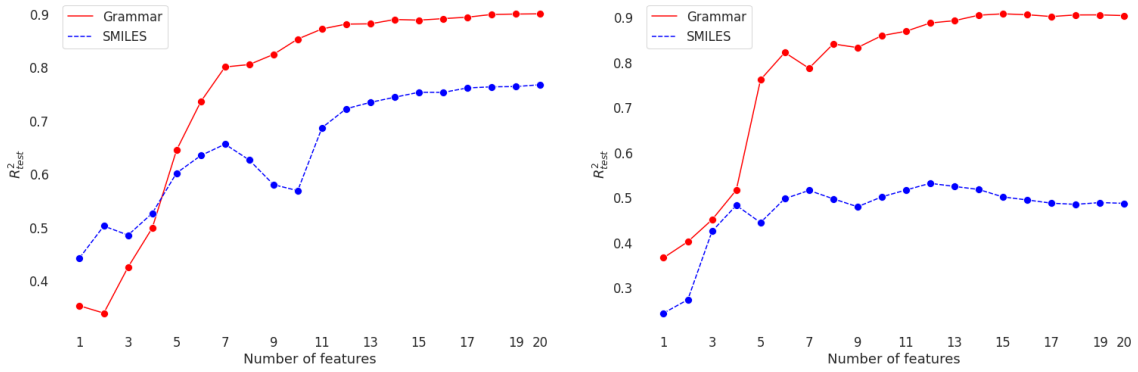


(b) T_b vs P_c

Figure 11: Comparison of feature importance (contributions) for regression models for T_b, T_c, and P_c. Based on the underlying chemistry-based correlations, it is expected that T_b and T_c would have similar feature importance, whereas T_b and P_c would have relatively higher differences in feature importance. This behavior is observed in the above comparison plots between T_b vs T_c and T_b vs P_c.

5.4. Model pruning

Now that we have information on the importance of various features for estimating molecular properties of interest, we leverage this information to further simplify our model by pruning the features. To do this, for each representation, we build models starting with a model with just the top-most important feature and compute the R^2_{test} value, then build another model with this feature and the next most important feature and again compute the R^2_{test} value, and do this sequentially until a sufficiently good performance is observed for the mode. The plot of the R^2_{test} values for sequentially developed models (from 1 to top-20 features) for both the representation vs the number of features is presented in Figures 12 (a) and (b) for T_b and T_c model.



(a) T_b (b) T_c

Figure 12: The R^2_{test} as a function of the number of features for the different models.

Table 3: Comparison of model performance (test-R^2) with different number of features.

	# Features	Grammar	SMILES
T_b	32	0.91	0.79
	20	0.90	0.77
T_c	32	0.90	0.48
	20	0.90	0.49

We observe that both the models require just 20 features or fewer to achieve nearly the same level of performance as obtained by using all the 32 features. The comparison of the performance on the test set using 32 and 20 features for T_b and T_c predictions are presented in Table 3. Therefore, the final pruned model is much simpler and is characterized by just 20 features that are dense, rich, and capture the molecular structural information efficiently. Moreover, we observe from the trend in Figure 12 that in both the cases, the grammar-based models achieve a significantly higher R^2 by using fewer features to attain the same accuracy both for T_b and T_c predictions. For example, the grammar model achieves an R^2 of nearly 0.8 for both T_b and T_c using just 7 features whereas it requires at least 20 features for the SMILES-based models to achieve a similar R^2. These again point toward the richness of the grammar features and its better generalization, as demonstrated earlier in the section information-theoretic analysis and complexity-agnostic nature of the model shown through the error vs. CSF analysis. In addition, fewer features would translate to better computational efficiency and a smoother decision boundary – both of which result in better generalizability of the model on unseen data points. In other words, the model predictions would have smaller variance and hence lower possibility of overfitting (or memorizing) the training data.

6. Conclusions

Thermodynamic property prediction is an important problem that requires a systematic approach to ensure the data-driven models are rooted properly in fundamental principles of physics and chemistry. Engineering the molecular representations that are input to such models in a way that maximizes the amount of physics/chemistry captured is one possible approach. Here, we have identified the SMILES grammar-based, dense vector representations obtained using the Grammar2vec framework to be one such promising possibility. These representations were obtained by invoking the natural language analogy to generate vector representations for 'molecular sentences' constructed by combining the SMILES string with their underlying SMILES grammar production rules.

The results demonstrate the superiority of the Grammar2vec molecular representations. They are shown to be associated with richer features that capture different chemistry, lower conditional uncertainty, better regression statistics both for data-rich (T_b) and data-limited (T_c) scenarios, higher efficiency in capturing molecular complexity indicated by the error analysis, and relatively simpler models with fewer features when compared to models using other representations. We have established that data alone is not sufficient to make better predictions – as is often assumed in the current era of deep learning – at least in chemistry. Instead, data along with richer representations that capture the intricacies of the underlying entities is of equal importance. The Grammar2vec framework could be used for estimating several other thermodynamic properties (in addition to T_b, T_c) using relatively simple and interpretable machine learning models.

We envision that such representations that are rooted in chemistry would be of significant value not only for the thermodynamic property prediction task but also for other chemistry problems requiring data-driven modeling. In future, we plan to extend this approach to incorporate additional molecular structure descriptors, utilize the grammar2vec framework to perform comprehensive property estimation on other thermodynamic properties with larger datasets, and perform molecule design and optimization in combination with retrosynthesis planning.

Appendix A

The SMILES grammar used in our work comprises 80 production rules with 24 non-terminals symbols specifying the different structural components of a SMILES string and are summarized in Table 4. The first and the last production rules, SMILES ⟶ CHAIN and NOTHING ⟶ NONE, are additional rules included signifying the start and end of a SMILES string, which is analogous to the <START> and <END> tokens in natural language processing marking the beginning and the end of sentences, respectively.

Table 4: SMILES grammar used in this work for generating molecular representations.

S.No	Production rules
1	SMILES ⟶ CHAIN
2	ATOM ⟶ BRACKET_ATOM | ALIPHATIC_ORGANIC | AROMATIC_ORGANIC
3	ALIPHATIC_ORGANIC ⟶ B | C | N | O | S | P | F | I | Cl | Br
4	AROMATIC_ORGANIC ⟶ c | n | o | s | p
5	BRACKET_ATOM ⟶ [BAI]
6	BAI ⟶ ISOTOPE SYMBOL BAC | SYMBOL BAC | ISOTOPE SYMBOL | SYMBOL
7	BAC ⟶ CHIRAL BAH | BAH | CHIRAL
8	BAH ⟶ HCOUNT BACH | BACH | HCOUNT
9	BACH ⟶ CHARGECLASS | CHARGE | CLASS
10	SYMBOL ⟶ ALIPHATIC_ORGANIC | AROMATIC_ORGANIC | ELEMENT_SYMBOLS
11	ISOTOPE ⟶ DIGIT | DIGIT DIGIT | DIGIT DIGIT DIGIT
12	DIGIT ⟶ 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
13	CHIRAL ⟶ @ | @@
14	HCOUNT ⟶ H | H DIGIT
15	CHARGE ⟶ - | - DIGIT | - DIGIT DIGIT | + | + DIGIT | + DIGIT DIGIT
16	BOND ⟶ - | = | # | / | \\
17	RINGBOND ⟶ DIGIT | BOND DIGIT
18	BRANCHED_ATOM ⟶ ATOM | ATOM RB | ATOM RB BB
19	RB ⟶ RB RINGBOND | RINGBOND
20	BB ⟶ BB BRANCH | BRANCH
21	BRANCH ⟶ (CHAIN) | (BOND CHAIN)
22	CHAIN ⟶ BRANCHED_ATOM | CHAIN BRANCHED_ATOM | CHAIN BOND BRANCHED_ATOM

Table 4: SMILES grammar used in this work for generating molecular representations.

S.No	Production rules
23	CLASS ⟶ DIGIT
24	ELEMENT_SYMBOLS ⟶ H
25	NOTHING ⟶ NONE

Appendix B

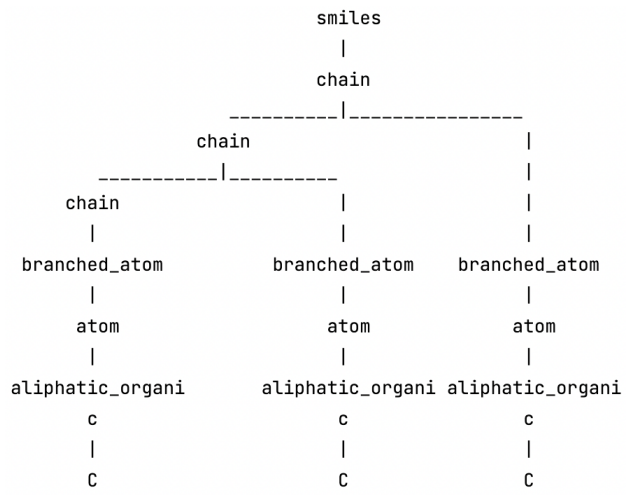


Figure 13: The parse-tree obtained for **propane** (CCC) using SMILES grammar productions in Table 1. The productions when extracted from the parse tree in a depth-first manner result in the **grammar representation** as 1, 2, 2, 4, 6, 15, 17, 6, 15, 17, 6, 15, 17

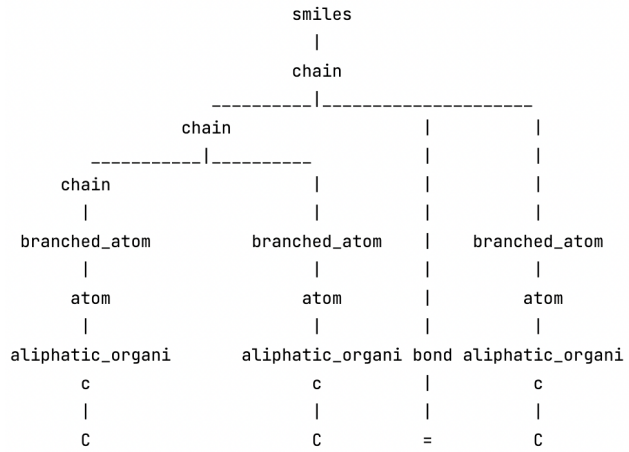


Figure 14: The parse-tree obtained for **propene** (CC=C) using SMILES grammar productions in Table 1. The productions when extracted from the parse tree in a depth-first manner result in the **grammar representation** as 1, 3, 2, 4, 6, 15, 17, 6, 15, 17, 13, 6, 15, 17

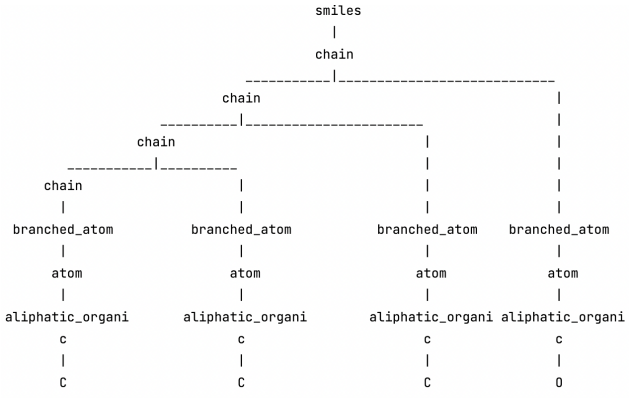


Figure 15: The parse-tree obtained for **propanol** (`CCCO`) using SMILES grammar productions in Table 1. The productions when extracted from the parse tree in a depth-first manner result in the **grammar representation** as 1, 2, 2, 2, 4, 6, 15, 17, 6, 15, 17, 6, 15, 17, 6, 15, 18

CCC	**C=CC**	**OCCC**
-0.491271	-0.698095	-0.434662
1.936522	2.444691	1.986346
2.12129	1.94433	2.406679
-1.778707	-2.328862	-1.935335
-0.91106	-0.799948	-0.95961
-2.156449	-1.863728	-1.858846
-2.249398	-2.12342	-1.562154
-0.147146	-0.343379	0.058143
-1.313123	-2.304307	-0.96131
-1.504984	-1.391153	-1.358496
-0.049938	0.295251	0.097997
0.576202	0.379481	0.391373
1.272612	1.822762	0.908829
1.422946	0.920623	1.252951
-2.824167	-2.334167	-2.607435
-0.862446	-0.855732	-0.668328
0.995295	0.973911	0.960458
1.48523	1.793601	1.430849
1.632852	2.427182	1.698906
-0.765502	-0.656052	-0.617677
-0.742146	-0.692423	-0.589372

0.835875	0.500569	0.396088
-4.488621	-4.736897	-4.288099
4.207507	4.423007	4.052883
-1.884889	-2.153435	-1.684574
-4.196161	-4.157901	-4.124537
2.292979	2.165808	2.419326
0.904267	0.696876	0.868011
1.223593	0.593412	1.06907
1.318167	1.215398	1.270406
-1.128037	-1.048277	-0.730391
0.08552	-0.223942	0.119815

Table 5: The dense, 32-dimensional representations obtained using the Grammar2vec framework on the grammar representation obtained using the complete grammar (provided in Appendix A) to parse the molecules' SMILES.

References

1. Venkatasubramanian, V. & Mann, V. Artificial intelligence in reaction prediction and chemical synthesis. *Current Opinion in Chemical Engineering* **36,** 100749 (2022).

2. Visco Jr, D. P., Pophale, R. S., Rintoul, M. D. & Faulon, J.-L. Developing a methodology for an inverse quantitative structure-activity relationship using the signature molecular descriptor. *Journal of Molecular Graphics and Modelling* **20,** 429–438 (2002).

3. Gani, R. Group contribution-based property estimation methods: advances and perspectives. *Current Opinion in Chemical Engineering* **23,** 184–196 (2019).

4. Lydersen, A. Estimation of critical properties of organic compounds. *Univ. Wisconsin Coll. Eng., Eng. Exp. Stn. Rep. 3* (1955).

5. Joback, K. G. & Reid, R. C. Estimation of pure-component properties from group-contributions. *Chemical Engineering Communications* **57,** 233–243 (1987).

6. Constantinou, L. & Gani, R. New group contribution method for estimating properties of pure compounds. *AIChE Journal* **40,** 1697–1710 (1994).

7. Benson, S. W. *et al.* Additivity rules for the estimation of thermochemical properties. *Chemical Reviews* **69,** 279–324 (1969).

8. Marrero, J. & Gani, R. Group-contribution based estimation of pure component properties. *Fluid Phase Equilibria* **183,** 183–208 (2001).

9. Hukkerikar, A. S. *et al.* Group-contribution+ (GC+) based estimation of properties of pure components: improved property estimation and uncertainty analysis. *Fluid Phase Equilibria* **321,** 25–43 (2012).

10. Gmehling, J. & Rasmussen, P. Vapor-liquid equilibria using UNIFAC. *AmsterdamB Elsevier* **151162** (1977).

11. Herraiz, J., Shen, S., Fernandez, J. & Coronas, A. Thermophysical properties of methanol+ some polyethylene glycol dimethyl ether by UNIFAC and DISQUAC group-contribution models for absorption heat pumps. *Fluid phase equilibria* **155,** 327–337 (1999).

12. Dhakal, P., Roese, S. N., Lucas, M. A. & Paluch, A. S. Predicting limiting activity coefficients and phase behavior from molecular structure: expanding MOSCED to alkanediols using group contribution methods and electronic structure calculations. *Journal of Chemical & Engineering Data* **63,** 2586–2598 (2018).

13. Tihic, A., Kontogeorgis, G. M., Von Solms, N., Michelsen, M. L. & Constantinou, L. A predictive group-contribution simplified PC-SAFT equation of state: application to polymer systems. *Industrial & Engineering Chemistry Research* **47,** 5092–5101 (2008).

14. Katritzky, A. R. *et al.* Quantitative correlation of physical and chemical properties with chemical structure: utility for prediction. *Chemical reviews* **110,** 5714–5789 (2010).

15. Gmehling, J., Constantinescu, D. & Schmid, B. Group contribution methods for phase equilibrium calculations. *Annual review of chemical and biomolecular engineering* **6,** 267–292 (2015).

16. Zhang, Z., Li, H., Chang, H., Pan, Z. & Luo, X. *Machine learning predictive framework for CO2 thermodynamic properties in solution. J CO2 Util 26: 152–159* 2018.

17. Yalamanchi, K. K. *et al.* Machine learning to predict standard enthalpy of formation of hydrocarbons. *The Journal of Physical Chemistry A* **123,** 8305–8313 (2019).

18. Saldana, D. A. *et al.* Prediction of density and viscosity of biofuel compounds using machine learning methods. *Energy & fuels* **26,** 2416–2426 (2012).

19. Alshehri, A. S., Tula, A. K., You, F. & Gani, R. Next generation pure component property estimation models: With and without machine learning techniques. *AIChE Journal,* e17469 (2021).

20. Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **35,** 1798–1828 (2013).

21. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling* **50,** 742–754 (2010).

22. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **28,** 31–36 (1988).

23. Yang, K. *et al.* Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling* **59,** 3370–3388 (2019).

24. Kusner, M. J., Paige, B. & Hernández-Lobato, J. M. *Grammar variational autoencoder* in *International Conference on Machine Learning* (2017), 1945–1954.

25. Mann, V. & Venkatasubramanian, V. Predicting chemical reaction outcomes: A grammar ontology-based transformer framework. *AIChE Journal* **67,** e17190 (2021).

26. Mann, V. & Venkatasubramanian, V. Retrosynthesis prediction using grammar-based neural machine translation: An information-theoretic approach. *Computers & Chemical Engineering* **155,** 107533. ISSN: 0098-1354 (2021).

27. Shapley, L. S. *A value for n-person games, Contributions to the Theory of Games, 2, 307–317* 1953.

28. Honda, S., Shi, S. & Ueda, H. R. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738* (2019).

29. Jaeger, S., Fulle, S. & Turk, S. Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling* **58,** 27–35 (2018).

30. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. *Distributed representations of words and phrases and their compositionality* in *Advances in neural information processing systems* (2013), 3111–3119.

31. Goh, G. B., Hodas, N. O., Siegel, C. & Vishnu, A. Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint arXiv:1712.02034* (2017).

32. Gong, Z., Wu, Y., Wu, L. & Sun, H. Predicting thermodynamic properties of alkanes by high-throughput force field simulation and machine learning. *Journal of chemical information and modeling* **58,** 2502–2516 (2018).

33. Pinheiro, G. A. *et al.* Machine learning prediction of nine molecular properties based on the SMILES representation of the QM9 quantum-chemistry dataset. *The Journal of Physical Chemistry A* **124,** 9854–9866 (2020).

34. Ding, J. *et al.* Machine learning for molecular thermodynamics. *Chinese Journal of Chemical Engineering* **31,** 227–239 (2021).

35. Aldosari, M. N., Yalamanchi, K. K., Gao, X. & Sarathy, S. M. Predicting entropy and heat capacity of hydrocarbons using machine learning. *Energy and AI* **4,** 100054 (2021).

36. Chomsky, N. On certain formal properties of grammars. *Information and control* **2,** 137–167 (1959).

37. Rehurek, R. & Sojka, P. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* **3** (2011).

38. Su, Y. *et al.* An architecture of deep learning in QSPR modeling for the prediction of critical properties using molecular signatures. *AIChE Journal* **65,** e16678 (2019).

39. Vapnik, V. N. The nature of statistical learning. *Theory* (1995).

40. Lundberg, S. M. & Lee, S.-I. *A unified approach to interpreting model predictions* in *Proceedings of the 31st international conference on neural information processing systems* (2017), 4768–4777.

41. Ghorbani, A. & Zou, J. *Data shapley: Equitable valuation of data for machine learning* in *International Conference on Machine Learning* (2019), 2242–2251.

42. Merrick, L. & Taly, A. *The explanation game: Explaining machine learning models using shapley values* in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (2020), 17–38.

43. Rodriguez-Perez, R. & Bajorath, J. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. *Journal of computer-aided molecular design* **34,** 1013–1026 (2020).

44. Smith, M. & Alvarez, F. Identifying mortality factors from Machine Learning using Shapley values–a case of COVID19. *Expert Systems with Applications* **176,** 114832 (2021).

45. Shannon, C. E. A mathematical theory of communications. *Bell Syst. Tech. J.* **27,** 379–423 (1948).

46. Szymkuć, S. *et al.* Computer-assisted synthetic planning: the end of the beginning. *Angewandte Chemie International Edition* **55,** 5904–5937 (2016).