Chemomile: An Explainable Multi-Level GNN Model for Combustion Property Prediction

Beomgyu Kang and Bong June Sung*

Department of Chemistry, Sogang University, Seoul 04172, Republic of Korea

E-mail: bjsung@sogang.ac.kr

Abstract

Accurate combustion property prediction is critical for safety since errors can lead to a hazardous situation. Many studies have focused on minimizing these errors, and recently models based on graph neural networks (GNNs) and message-passing neural networks (MPNNs) have shown promise. However, these models often neglect the hierarchical structure and 3D geometries of atoms within molecules. This study introduces Chemomile, a novel explainable geometry-based GNN model specifically designed for combustion property prediction. Chemomile is optimized using particle swarm optimization (PSO) and benchmarked against five key combustion properties. Chemomile shows competitive performance with existing GNN models. Using a perturbation-based explanation method, atom-wise contribution can be quantified, providing valuable insights into how each atom influences the overall combustion property. This information aligns with existing knowledge and can be a powerful tool for chemists designing molecules with specific combustion behaviors.

Keywords

Graph Neural Network, QSPR, Machine Learning, Combustion Property, Explainable GNN

Introduction

Accurate prediction of combustion properties is crucial to ensure safety. Errors in these predictions can lead to serious safety hazards. Furthermore, identifying properties solely for existing compounds is insufficient since new compounds can be formed during combustion. This highlights the need for models that can handle the dynamic nature of the combustion process. For this reason, many existing models have tried to predict combustion properties accurately without experiments. These models are based on various methods, such as correlating the number of atoms in a molecule to properties[1,2] and quantifying the effect of certain types of functional groups.[3,4] While these methods offer valuable insights, they may not capture the complex interplay between factors influencing combustion properties. To address these limitations, machine learning (ML) based on quantitative structure-property relationship (QSPR) has been introduced.[5–9] These models use molecular descriptors to identify the complex link between a molecule's structure and properties. Beyond improved prediction accuracy, ML approaches can also provide valuable chemical insights into the combustion process, previously unknown to chemists.

Graph neural networks (GNNs) have emerged as a powerful tool for predicting the properties of molecules. This approach is based on the inherent graph structure of a molecule, where atoms are represented as nodes (V) and chemical bonds are represented as edges (E) connecting these nodes. Many recent studies have focused on developing methods to extract more detailed information from this molecular graph representation to achieve accurate property predictions. A key concept in GNNs is the idea of a message-passing neural network (MPNN). In an MPNN, the hidden state of a node (representing an atom) is updated iteratively. This process can be interpreted as an atom exchanging information with its neighboring atoms through the connected bonds, mimicking how atoms influence each other in a molecule. Building upon this idea, researchers have introduced various GNN models to predict diverse properties of a molecule, such as quantum chemical properties,[10] an activity to certain enzymes,[11] and the protein-ligand binding affinities.[12]

Fragmenting a molecule into smaller substructures to capture more detailed information has gained traction recently. This approach allows GNNs to learn from a molecule's overall structure and constituent parts. In 2020, Zhang et al. introduced FraGAT, a GNN architecture that utilizes fragmentation.[13] FraGAT achieved this by breaking down a molecule at acyclic single bonds, generating a set of individual fragments. The model then leverages information about how these fragment pairs interact. Each fragment pair can make a prediction, and these multiple predictions are subsequently averaged to provide a final prediction for the entire molecule. Building upon this idea, Aouichaoui et al. proposed GroupGAT.[14,15] Instead of relying on generic fragment discovery, GroupGAT leverages pre-defined functional groups derived from group contribution (GC) studies. This approach allows GroupGAT to achieve state-of-the-art accuracy on various combustion property prediction tasks. While fragmentation offers valuable insights, it's important to acknowledge that the 3D spatial arrangement of atoms within a molecule also plays a crucial role in its properties. While some fragmentation methods may implicitly capture some aspects of 3D structure through fragment interactions, ongoing research is needed to integrate explicit 3D information into GNN models for molecule representation.

Beyond achieving accurate property predictions, understanding the rationale behind those predictions is crucial. In 2023, Wu et al. proposed a method for explaining GNN prections called substructure mask explanation (SME).[16] The core idea of SME revolves around masking specific substructures within a molecule. By masking these substructures, the model's prediction changes, and the difference between the original and masked predictions reveals the contribution of the masked substructure to the overall property. The author applied SME to measure the contribution of different chemical substructures to a molecule's water solubility. The obtained results largely aligned with established chemical knowledge. This approach can be valuable in designing new compounds with desired properties. Since SME is built upon the concept of fragmentation, it primarily explains the contribution of individual substructures. However, the interplay between various substructures and their

spatial arrangement within the molecule can be crucial for complex properties.

This work introduces Chemomile, a novel explainable GNN model designed for predicting combustion properties of molecules. Chemomile utilizes geometric information alongside graph-based representations to achieve accurate predictions. Chemomile employs a multi-level graph representation of a molecule: a molecule-level, several fragment-level, and a junction-tree level. Crucially, Chemomile incorporates the optimized geometry for each substructure into representations at each level. This allows the model to capture the atomic composition and spatial arrangement of atoms within the molecule. Each level of the graph representation undergoes a repetitive update process using a graph attention mechanism. This mechanism helps the model focus on the most relevant parts of the graph for the prediction task. The resulting embeddings from each level are then processed to construct a final molecular embedding that encapsulates the key features of the molecule. Chemomile is benchmarked on its ability to predict various combustion properties. The model's hyperparameters are optimized using particle swarm optimization (PSO) to ensure optimal performance. One of Chemomile's strengths lies in its explainability. A perturbation-based explanation algorithm allows for the quantification of atom-wise contributions to the predicted target properties. This enables us to understand how each atom in a molecule influences the overall combustion behavior. Chemomile demonstrates competitive performance across different target combustion properties. Additionally, the interpretations obtained from the atom-wise contribution analysis align with established chemical knowledge, highlighting the model's ability to provide meaningful insights.

Methods

Fragmentation and featurization

When acyclic single bonds in a molecule are broken, several fragments can be identified. This fragment concept is used in FraGAT.[13] We can represent a molecule as a graph $G = \{V,$

$E\}$ where V and E represent atoms and bonds, respectively, multi-level fragmental graphs can be obtained; one for the entire molecule and others for each fragment. By applying this concept, we can obtain multi-level fragmental graphs: one for the entire molecule and others for each fragment. The fragmentation process is illustrated in Figure 1. Then, the atomic geometry of each fragment (including the entire molecule) can be optimized using MMFF force field,[17,18] producing coordinates for each atom. The features of atoms and bonds in each fragment can be extracted now. The list of atomic and bond features used in this study is listed in Table 1.

Table 1: List of features used for atom and bond featurization

Atom Feature	Description	Length
atomic number	atomic number of an atom	1
chirality	whether the atom is chiral center	1
degree	number of covalent bonds of an atom	1
formal change	formal charge of an atom	1
hydrogens	number of hydrogens connected to an atom	1
radical electrons	number of radical hydrogens of an atom	1
hybridization	hybridization type (sp^3, sp^2, etc.) of an atom	1
aromatic	whether an atom is in an aromatic system	1
ring	whether an atom is in a ring	1
coordinate	relative coordinate from the center of the fragment (x, y, z)	3
Bond Feature	**Description**	**Length**
type	type (single, aromatic, etc.) of a bond	1
stereo	type of stereo (E/Z, none, any) of a bond	1
conjugation	whether a bond is conjugated	1

We can also consider an additional level of graph representation, where nodes and edges represent fragments and the connection between fragments, respectively. This structure is called a junction tree. In total, as shown in Figure 3, this approach allows us to obtain three levels of graphs from a single molecule; a molecule-level graph, multiple fragment-level graphs, and a junction tree. This information will be used later to create a model that can predict certain properties of molecules.

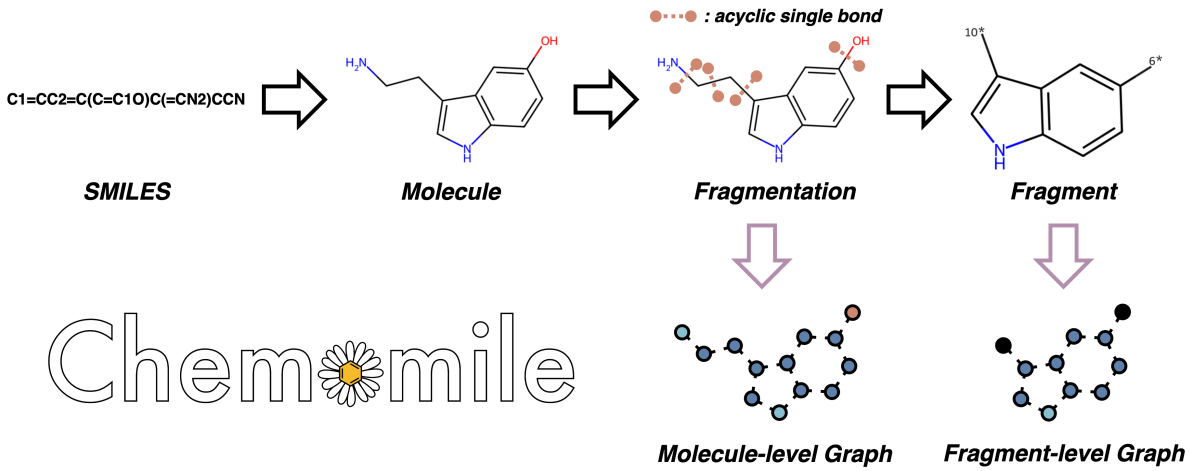

Figure 1: Schemetic picture of molecule fragmentation. A molecule identified the by SMILES sequence can be broken into several smaller pieces, called fragments, by breaking acyclic single bonds. Both the original molecule and each fragment can be represented as a graph. Here, atoms and bonds are represented by nodes and edges, respectively.

AttentiveFP

In 2020, Xiong et al. introduced a new approach to molecular representation with a GNN model named AttentiveFP.[19] This model leverages a graph attention mechanism to capture property-relevant features of a molecule.

Figure 2 shows the two main phases of AttentiveFP.

Phase 1: Atom Embedding Update

- Alignment: This step calculates a vector (e_{vu}) representing the compatibility between an atom (v) and its neighbor (u). This considers the current representation of both atoms (h_v and h_u) using a learnable weight matrix (W). (1)
- Weighting: A softmax function is applied to the alignment vector (e_{vu}), resulting in an attention score (a_{vu}). This score indicates the relative importance of neighbor u for atom v. (2) Here, $N(v)$ represents the set of neighboring atoms for atom v.
- Contest Generation: A weighted sum of the neighbor representations (h_u) is calculated, considering the attention scores (a_{vu}). This weighted sum is then passed

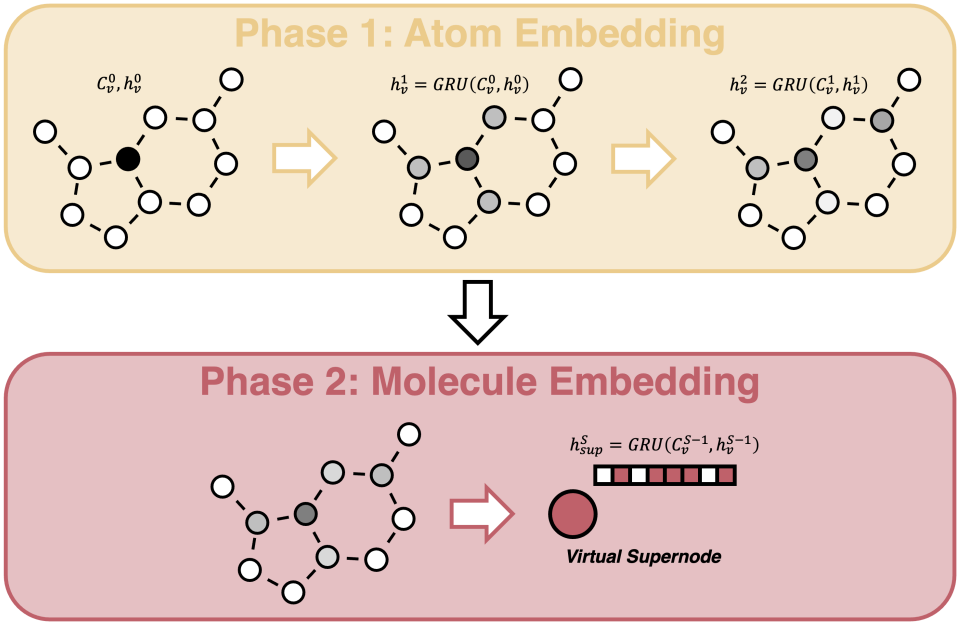

Figure 2: Schemetic picture of the two main stages of AttentiveFP: atom embedding and molecule embedding. In the iterative atom embedding phase (top), information from a single atom (circle) is used to update the representation of its neighbors (grey). The molecule embedding phase (bottom) leverages the attention mechanism to create a single embedding for the entire molecule onto a virtual supernode by focusing on the most influential features captured during the atom embedding updates. This approach allows AttentiveFP to capture property-relevant information from a molecule for various applications.

through an *elu* activation function to obtain a context vector (C_v). This contest vector captures the influence of surrounding atoms on atom v. (3)

- State Update: Finally, a Gated Recurrent Unit (GRU) leverages the previous state of atom v (h_v^{k-1}) and the contest vector (C_v^{k-1}) to update its current representation (h_v^k). This iterative process ensures information propagation across neighboring atoms. (4)

Phase 2: Molecule Embedding The second phase utilizes a similar attention mechanism but on a virtual supernode (v_{sup}). This supernode essentially holds the combined representation of the entire molecule after the atom embedding updates. Here, another GRU updates the supernode representation ($h_{v_{sup}}^s$) based on its previous state ($h_{v_{sup}}^{s-1}$) and a contest vector ($C_{v_{sup}}^{s-1}$). (5)

This two-phase approach allows AttentiveFP to capture the relationships between atoms and ultimately create a molecule-level representation that considers the most relevant atomic features.

$$e_{vu} = leaky\_relu(W \cdot [h_v, h_u]) \tag{1}$$

$$a_{vu} = softmax(e_{vu}) = \frac{exp(e_{vu})}{\sum_{u \in N(v)} exp(e_{vu})} \tag{2}$$

$$C_v = elu(\sum_{u \in N(v)} a_{vu} \cdot W \cdot h_u) \tag{3}$$

$$h_v^k = GRU^{k-1}(C_v^{k-1}, h_v^{k-1}) \tag{4}$$

$$h_{v_{sup}}^s = GRU^{s-1}(C_{v_{sup}}^{s-1}, h_{v_{sup}}^{s-1}) \tag{5}$$

Chemomile Network

As shown in Figure 3, the Chemomile network leverages molecule fragmentation to create a rich representation for property prediction. After processing a molecule, three graphs are obtained: molecule-level, fragment-level (one for each fragment), and junction-tree-level. The molecule-level graph is fed into an AttentiveFP layer (AFP_MOL) to generate a molecule embedding (shown in blue). Similarly, each fragment-level graph is processed by another attentiveFP layer (AFP_FRAG) to create individual fragment embeddings (shown in green). An embedding is also obtained from (AFP_JT) (shown in orange). Finally, the molecule embedding and the junction tree embedding are concatenated. This combined representation is then fed to a fully connected layer (FCL) to make the final prediction about the molecule's properties.

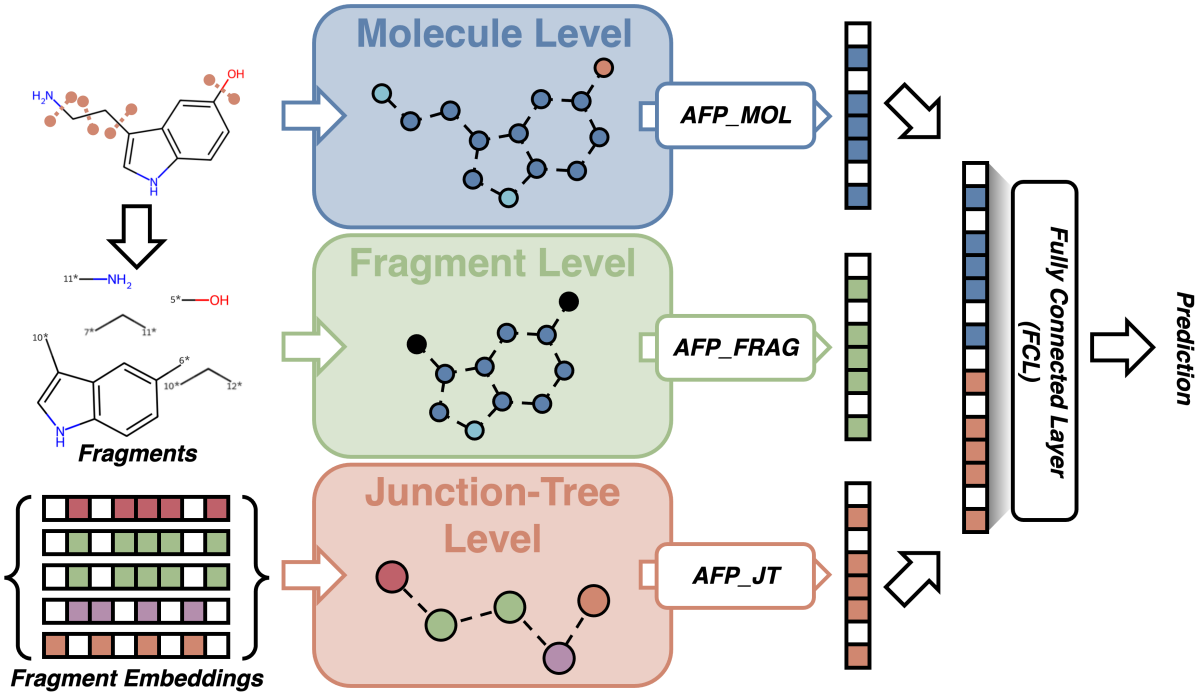

Figure 3: Schemetic picture of the workflow of Chemomile network. The Chemomile network workflow starts with a molecular structure. Fragmentation breaks the molecule into smaller parts, and both the whole and its fragments are represented as graphs. A junction tree connects these fragments. Finally, embeddings are created at each level and potentially combined to represent the entire molecule.

Particle Swarm Optimization

PSO is used for hyperparameter searching since it is more adaptable for parallelization than iterative methods such as Bayesian Optimization (BO). PSO is based on mimicking the social behavior of animals, where a set of candidate solutions explore the search space for an optimal solution (minimum error in this case).

1. Initialization: A set of candidate solutions (particles) are distributed through the hyperparameter space. Each particle represents a potential configuration of hyperparameters for the model.

2. Velocity Update: Each particle considered three factors to determine its movement in the next iteration.

    - Personal Best: The particle's own best position encountered so far.
    - Social Best: The best position discovered by any particle in the swarm.
    - Inertia: A tendency to continue moving in the same direction as previous iterations.

3. Iteration: These steps are repeated for a predetermined number of iterations, allowing the particles to explore the search space and converge toward the optimal solution.

Figure 4 illustrates how this iterative process can lead to a global minimum point for error.

PySwarm[20] is used to implement PSO for hyperparameter search. The search involves 10 candidate solutions exploring the hyperparameter space for 50 iterations. This translated to 500 total training and testing cycles for the model during the search process. The goal is to find the combination of hyperparameters that minimizes the test set loss. Each model's performance is evaluated using this metric. The weights for the three factors influencing particle movement in PSO are set to:

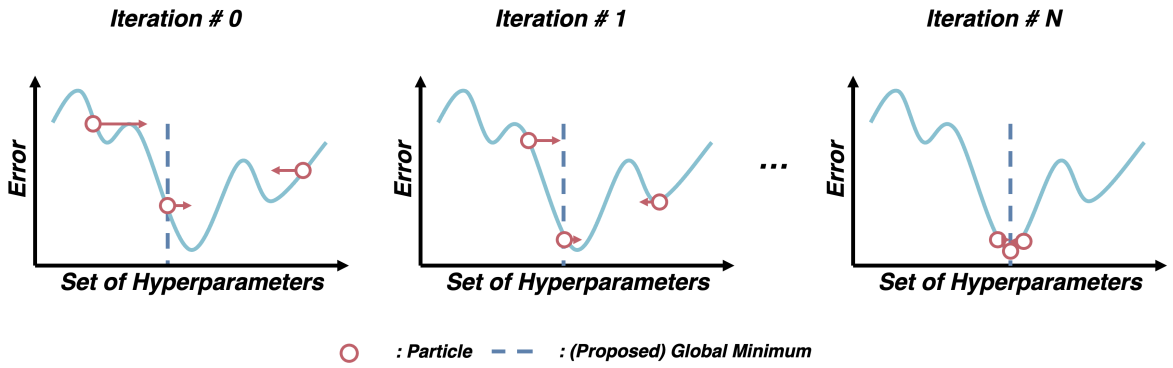

Figure 4: Schemetic picture of particle swarm optimization (PSO). PSO uses particles (circles) to explore a search space for optimal solutions (minimum error). Particles move based on their own best position found so far, the best position found by any particle, and a tendency to keep moving in the same direction. This iterative helps them converge towards the best solution.

1. Independent Velocity: 0.5 (weight given to a particle's own best solution)

2. Social Velocity: 0.3 (weight given to the best position found by any particle)

3. Inertia: 0.9 (weight given to the particle's tendency to continue in its current direction).

Table 2 lists the specific hyperparameters being optimized, their description,ons, and the search space boundaries for each.

Table 2: List of hyperparameters for particle swarm optimization

hyperparameter	description	boundary
hidden_size	size of hidden layers	[10, 100]
dropout	dropout rate	[0.2, 0.5]
num_layers	number of GNN layers for graph attention mechanism	[1, 5]
num_timesteps	number of iterative steps for graph featurization	[1, 5]
lr_init	initial point for learning rate	$[10^{-5}, 10^{-1}]$
gamma	rate for exponential learning rate scheduler	[0.975, 0.999]
weight_decay	weight decay (L2 norm)	[0, 0.1]

Dataset

Our model is trained and tested on data from the Design Institute for Physical Property Research (DIPPR) 801 database, managed by the American Institute of Chemical Engineers

(AIChE). This database contains pairs of SMILES strings and corresponding property values for various compounds. During data preprocessing, we excluded compounds that failed the geometry optimization of their fragments, a crucial step in our approach. Additionally, we only selected compounds with experimentally determined properties that DIPPR has accepted to ensure data reliability. Table 3 lists the specific properties (targets) used for training and testing our model, along with the number of data points available for each property.

Table 3: Summary of combustion properties used in this study

Combustion Property	Unit	Number of data
Flashpoint (FP)	K	943
Autoignition temperature (AIT)	K	554
Enthalpy of combustion at 298.15 K (HCOM)	kJ mol^{-1}	876
Upper flammability limit (FLVU)	Vol % in air	397
Lower flammability limit (FLVL)	Vol % in air	465

Figure 5 illustrates the distribution of the target properties in our dataset. To stabilize the training process for Chemomikle, which predicts Z-scores, all data points are transformed into (SMILES, Z-score) pairs. The Z-score, defined by (6), standardizes the target value (property) by subtraction the mean ($\bar{X}$) of the entire dataset and then dividing by the standard deviation (s). This transformation normalizes the data distribution, making it easier for the model to learn during training. After the model predicts a Z-score, the original target value of the property can be recovered using the equation $X_{pred} = Z_{pred} \times s + \bar{X}$. This process reverses the Z-score transformation applied during data preprocessing. We use the original mean and standard deviation from the entire dataset to convert the predicted Z-score back to the corresponding property value on the original scale.

$$Z = \frac{X - \bar{X}}{s} \tag{6}$$

A common data splitting strategy is employed, where the dataset is partitioned into three subsets: 80% for training the model, 10% for validation during training, and the remaining

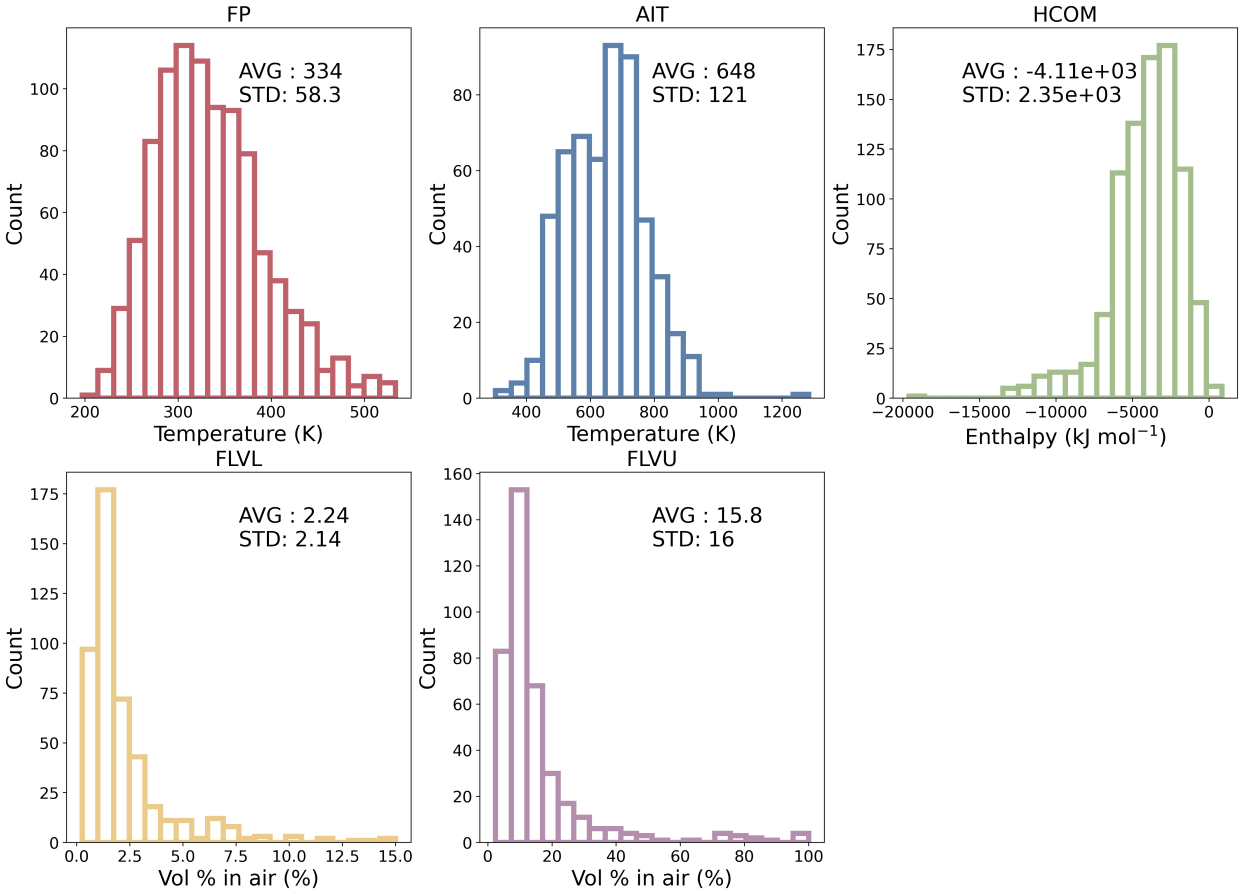

Figure 5: Distribution of data used in this work.

10% for final testing of the model's performance.

Training and Metrics

Chemomile is implemented using PyTorch[21] and PyG[22] libraries. Since the model aims for property regression, the mean squared error (MSE) loss is used as the objective function for all tasks. The Adam optimizer is employed for gradient descent, along with an exponential learning rate scheduler. Both the initial learning rate and the decay constant are optimized through hyperparameter search. Training is conducted for a maximum of 200 epochs. During training, the model's performance is monitored on a validation set. The model with the lowest validation loss is saved, and its performance on the held-out test set is evaluated using various error metrics. These metrics allow for a comprehensive comparison with other models. The error metrics used are:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{8}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left(|y_i - \hat{y}_i| \times \frac{100}{y_i}\right) \tag{9}$$

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

y_i, $\hat{y}_i$, $\tilde{y}$ and N means the true value of the ith data point, predicted value of the ith data point, mean of the true values in the test set, and total number of data points in the test set, respectively.

Results and discussion

Particle Swarm Optimization

PSO was employed to optimize the hyperparameters of Chemomile. The search involved 10 "agents" (candidate solutions) exploring the hyperparameter space for 50 iterations. The optimized hyperparameters for each target property are listed in Table 4.

Table 4: Optimized hyperparameters for Chemomile

Target	hidden_size	dropout	num_layers	num_timesteps	lr_init	gamma	weight_decay
FP	84	0.360	4	4	10^{-2}	0.994	3.5×10^{-3}
AIT	65	0.266	4	4	10^{-2}	0.995	1.8×10^{-3}
HCOM	64	0.435	4	4	10^{-2}	0.995	2.7×10^{-3}
FLVL	63	0.362	3	3	10^{-2}	0.982	1.3×10^{-2}
FLVU	68	0.335	2	3	10^{-3}	0.995	1.2×10^{-3}

As observed in the table, the optimal hyperparameter configuration varies for each target property. This suggests that the origin and influence of these properties on a molecule can differ. The obtained hyperparameters are then used in the subsequent benchmarking process to evaluate Chemomile's performance on various combustion properties.

Benchmark

The trained Chemomile with optimized hyperparameters was evaluated on both the entire dataset and the test set. Figure 6 visualizes the model's prediction accuracy, plotting actual versus predicted values. Error metrics (MAE, RMSE, MAPE, R^2) are included for both datasets within the figure. The detailed error metrics are provided in Table 5.

Chemomile achieves reasonable performance on most targets, with errors generally lower on the total dataset compared to the test set. This is expected as the model is trained on the total data and may slightly overfit the training data. Looking closer at Table 5, we see variations in performance across different properties. Chemomile achieves the best results (lowest errors and highest R^2) in HCOM, followed by FP and FLVL. The performance for

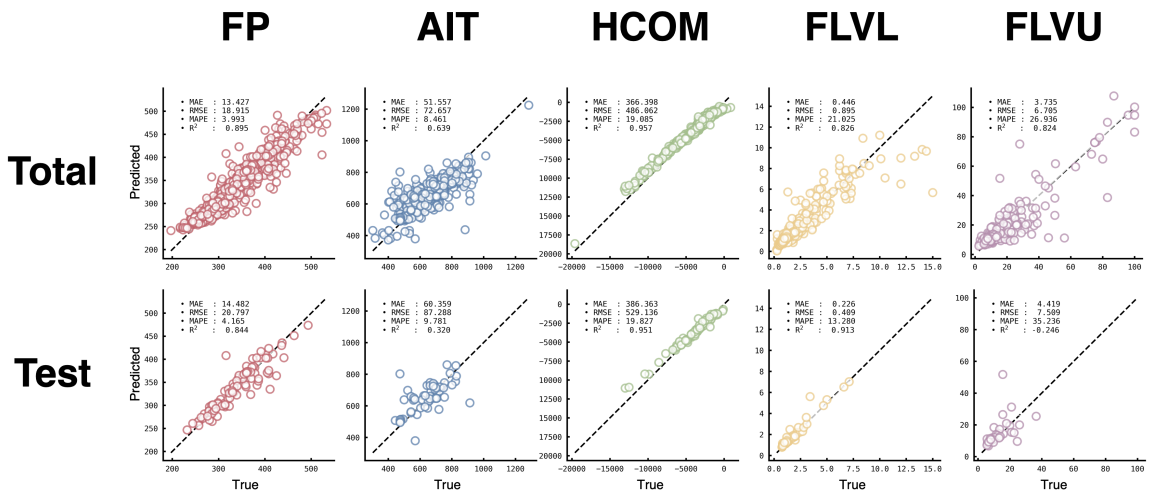

Figure 6: The actual vs. predicted values of Chemomile. The model is tested on the test set and the entire dataset. Error metrics for each result are included in the figure.

Table 5: Benchmark results of Chemomile. The performance of Chemomile on target combustion properties (FP, AIT, HCOM, FLVL, FLVU) is represented using error metrics (MAE, RMSE, MAPE, R^2). The results are shown for both the entire dataset and the test set.

Target	Total				Test			
	MAE (Unit)	RMSE (Unit)	MAPE (%)	R^2	MAE (Unit)	RMSE (Unit)	MAPE (%)	R^2
FP	13.427 (K)	18.915 (K)	3.993	0.895	14.482 (K)	20.797 (K)	4.165	0.844
AIT	51.557 (K)	72.567 (K)	8.461	0.639	60.359 (K)	87.288 (K)	9.781	0.320
HCOM	366.398 (kJ mol^{-1})	486.062 (kJ mol^{-1})	19.085	0.957	386.363 (kJ mol^{-1})	529.136 (kJ mol^{-1})	19.827	0.951
FLVL	0.446 (%)	0.895 (%)	21.025	0.826	0.226 (%)	0.409 (%)	13.280	0.913
FLVU	3.375 (%)	6.705 (%)	26.936	0.824	4.419 (%)	7.509 (%)	35.236	-0.246

AIT and FLVU is lower, particularly on the test set (indicated by a higher drop in R^2 compared to other targets.

To gain a broader understanding of Chemomile's effectiveness, its performance is compared with existing GNN models on each target property. Table 6 summarizes these comparisons, including references, model types, data sizes, and the specific error metric used by each study.

Table 6: Comparision of errors of Chemomile with other recent GNN models. (Abbreviations: GC - Group Contribution, ANN - Artificial Neural Network, MLR - Multi-Linear Regression, NLI - Non-Linear Interactions, DNN - Deep Neural Network)

Target (Unit)	Reference	Type	# data	Metric
FP (K)	Current Work	GNN	943	MAE : 13.427
	Aouichaoui et al. (2023)[14]	GNN	888	MAE : 3.75
	Sun et al. (2020)[23]	GNN	10,575	MAE : 17.8
AIT (K)	Current Work	GNN	554	MAE : 51.557
	Aouichaoui et al. (2023)[14]	GNN	888	MAE : 29.70
	Yang et al. (2021)[24]	DNN	888	MAE : 35.57
HCOM (kJ mol^{-1})	Current Work	GNN	876	MAE : 366.398
	Aouichaoui et al. (2023)[14]	GNN	853	MAE : 38.52
	Cao et al. (2009)[4]	GC + ANN	1,496	MAE : 155.40
FLVL (%)	Current Work	GNN	397	MAE : 0.446
	Aouichaoui et al. (2023)[14]	GNN	432	MAE : 0.17
	Charagheizi et al. (2008)[25]	QSPR+MLR	1,056	MAE : 7.68
FLVU (%)	Current Work	GNN	465	MAE : 3.375
	Aouichaoui et al. (2023)[14]	GNN	367	MAE : 1.66
	Park et al. (2021)[1]	QSPR+MLR+NLI	1,711	MAE : 2.44

Chemomile demonstrates competitive performance on most targets. In HCOM, Chemomile achieves lower errors than previously reported models. While Chemomile performs well, there is still room for improvement, particularly for targets like AIT and FLVU. Future work may explore different GNN architectures, data augmentation techniques, or hyperparameter tuning strategies to enhance performance further.

Explanation

A perturbation-based explanation method is used to quantify the contribution of each atom in the 5-hydroxytryptamine (serotonin) molecule to the predicted target properties.

Chemomile outputs Z-scores for each property and the contribution is measured by how much an atom's absence would shift the predicted Z-score, as shown in Figure 7.

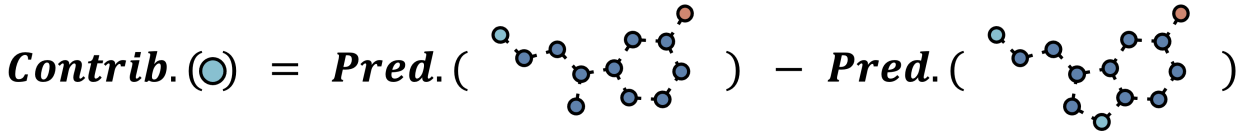

Figure 7: Schemetic picture of the quantifying contribution of each atom in a molecule. A contribution is defined as how much does the output changes in the absence of each atom.

Higher positive values indicate the atom contributes to a higher predicted value for that property, while lower negative values indicate the opposite. Figure 8 summarizes the contribution of atoms for predicting each target property.

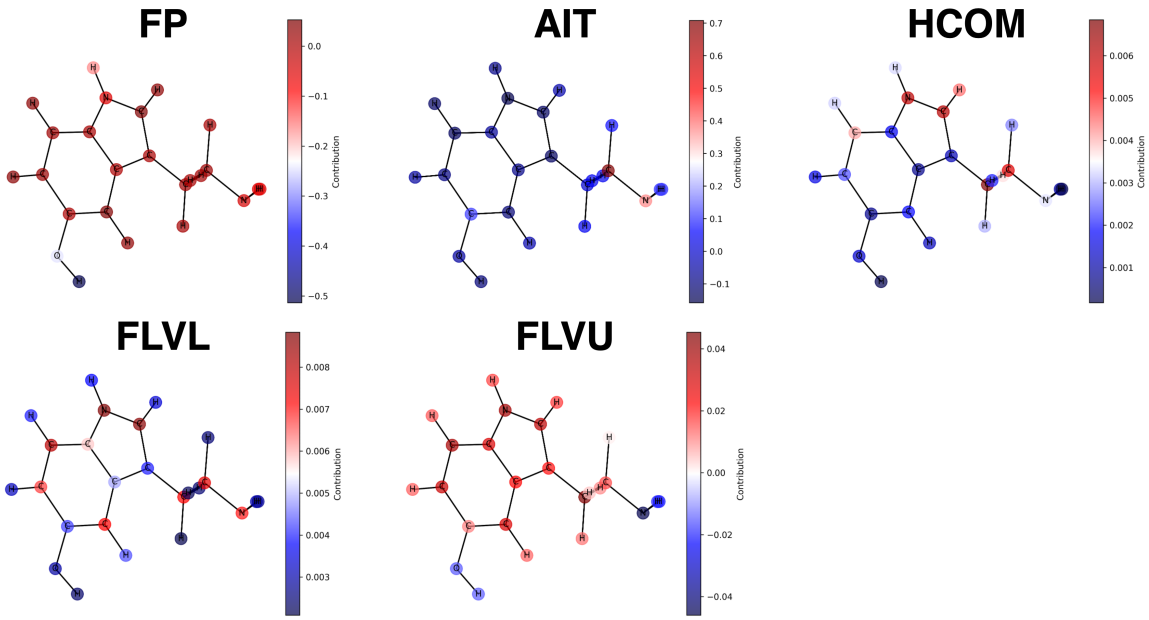

Figure 8: Atom-wise contribution to the predicted combustion properties for 5- hydroxytryptamine (serotonin). Higher positive values indicate the atom contributes to a higher predicted value for that property. The coordinates of atoms are obtained from geometry optimization results.

The figure showcases several key points regarding how individual atoms contribute to the predicted combustion properties of serotonin. As expected, the contribution of each atom

varies across different target properties. This highlights the unique relationships between an atom's presence and each specific combustion property. It's important to note that the explanation method relies on the accuracy of the underlying model. As a result, the observed range of atom contributions may differ for each target property. For example, AIT, which has the largest MAPE among target properties, also exhibits the largest range of atom contribution values. Conversely, properties like HCOM and FLVL show a smaller range of variation, potentially indicating more consistent model performance for these targets. These results also reveal the significant contributions of functional groups like hydroxy ($-OH$) and amino ($-NH_2$) groups to the predicted properties. This aligns with our existing chemical knowledge about how these groups can influence combustion behavior.

Conclusion

This work introduces Chemomile, a geometry-based multi-level GNN architecture for predicting the combustion properties of molecules. Chemomile leverages both the fragmentation of molecules and their atomic geometry to capture comprehensive information for accurate prediction. This approach allows the model to handle the dynamic nature of combustion processes. Chemomile was tested on five targets; FP, AIT, HCOM, FLVL, and FLVU. A model for each target is trained with the optimized hyperparameters obtained from PSO. The benchmark results demonstrate that Chemomile achieves competitive performance on most target properties compared to existing GNN models. Beyond prediction accuracy, Chemomile's explainability provides valuable insights into the factors influencing combustion properties. The atom-wise contribution analysis allows us to understand how individual atoms and functional groups contribute to the overall behavior of a molecule. This knowledge can be instrumental in the development of safer combustion processes. Overall, Chemomile offers a promising approach for combustion property prediction, with the potential to become a valuable tool for researchers and engineers in this field. Continued development and

refinement can enhance its accuracy, explainability, and applicability to a wider range of combustion problems.

Acknowledgement

This work was supported by Korea Environment Industry & Technology Institute(KEITI) through "Advanced Technology Development Project for Predicting and Preventing Chemical Accidents" Program, funded by Korea Ministry of Environment(MOE) (RS-2023-00219144)

References

(1) Park, S.; Bailey, J. P.; Pasman, H. J.; Wang, Q.; El-Halwagi, M. M. Fast, easy-to-use, machine learning-developed models of prediction of flash point, heat of combustion, and lower and upper flammability limits for inherently safer design. *Computers & Chemical Engineering* **2021**, *155*, 107524.

(2) Keshavarz, M. H.; Hasani, R. Reliable predictions of the net heat of combustion and the condensed phase heat of formation of organosilicon compounds. *Fuel* **2022**, *307*, 121931.

(3) Alibakhshi, A.; Mirshahvalad, H.; Alibakhshi, S. Prediction of flash points of pure organic compounds: Evaluation of the DIPPR database. *Process Safety and Environmental Protection* **2017**, *105*, 127–133.

(4) Cao, H.; Jiang, J.; Pan, Y.; Wang, R.; Cui, Y. Prediction of the net heat of combustion of organic compounds based on atom-type electrotopological state indices. *Journal of Loss Prevention in the Process Industries* **2009**, *22*, 222–227.

(5) Mirshahvalad, H.; Ghasemiasl, R.; Raoufi, N.; Malekzadeh Dirin, M. A neural network

QSPR model for accurate prediction of flash point of pure hydrocarbons. *Molecular Informatics* **2019**, *38*, 1800094.

(6) Borhani, T. N. G.; Afzali, A.; Bagheri, M. QSPR estimation of the auto-ignition temperature for pure hydrocarbons. *Process Safety and Environmental Protection* **2016**, *103*, 115–125.

(7) Dashti, A.; Jokar, M.; Amirkhani, F.; Mohammadi, A. H. Quantitative structure property relationship schemes for estimation of autoignition temperatures of organic compounds. *Journal of Molecular Liquids* **2020**, *300*, 111797.

(8) Dashti, A.; Mazaheri, O.; Amirkhani, F.; Mohammadi, A. H. Molecular descriptors-based models for estimating net heat of combustion of chemical compounds. *Energy* **2021**, *217*, 119292.

(9) Pan, Y.; Jiang, J.; Wang, R.; Cao, H.; Cui, Y. A novel QSPR model for prediction of lower flammability limits of organic compounds based on support vector machine. *Journal of hazardous materials* **2009**, *168*, 962–969.

(10) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. International conference on machine learning. 2017; pp 1263–1272.

(11) Kong, Y.; Zhao, X.; Liu, R.; Yang, Z.; Yin, H.; Zhao, B.; Wang, J.; Qin, B.; Yan, A. Integrating concept of pharmacophore with graph neural networks for chemical property prediction and interpretation. *Journal of Cheminformatics* **2022**, *14*, 52.

(12) Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P. graphDelta: MPNN scoring function for the affinity prediction of protein–ligand complexes. *ACS omega* **2020**, *5*, 5150–5159.

(13) Zhang, Z.; Guan, J.; Zhou, S. FraGAT: a fragment-oriented multi-scale graph attention model for molecular property prediction. *Bioinformatics* **2021**, *37*, 2981–2987.

(14) Aouichaoui, A. R.; Fan, F.; Abildskov, J.; Sin, G. Application of interpretable group-embedded graph neural networks for pure compound properties. *Computers & Chemical Engineering* **2023**, *176*, 108291.

(15) Aouichaoui, A. R.; Fan, F.; Mansouri, S. S.; Abildskov, J.; Sin, G. Combining Group-Contribution concept and graph neural networks toward interpretable molecular property models. *Journal of Chemical Information and Modeling* **2023**, *63*, 725–744.

(16) Wu, Z.; Wang, J.; Du, H.; Jiang, D.; Kang, Y.; Li, D.; Pan, P.; Deng, Y.; Cao, D.; Hsieh, C.-Y.; others Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature Communications* **2023**, *14*, 2585.

(17) Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of computational chemistry* **1996**, *17*, 490–519.

(18) Tosco, P.; Stiefl, N.; Landrum, G. Bringing the MMFF force field to the RDKit: implementation and validation. *Journal of cheminformatics* **2014**, *6*, 1–4.

(19) Xiong, Z.; Wang, D.; Liu, X.; Zhong, F.; Wan, X.; Li, X.; Li, Z.; Luo, X.; Chen, K.; Jiang, H.; others Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry* **2019**, *63*, 8749–8760.

(20) Miranda, L. J. V. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software* **2018**, *3*.

(21) Ansel, J. et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. 29th ACM International Conference on

Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24). 2024.

(22) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds. 2019.

(23) Sun, X.; Krakauer, N. J.; Politowicz, A.; Chen, W.-T.; Li, Q.; Li, Z.; Shao, X.; Sunaryo, A.; Shen, M.; Wang, J.; others Assessing graph-based deep learning models for predicting flash point. *Molecular Informatics* **2020**, *39*, 1900101.

(24) Yang, A.; Su, Y.; Wang, Z.; Jin, S.; Ren, J.; Zhang, X.; Shen, W.; Clark, J. H. A multi-task deep learning neural network for predicting flammability-related properties from molecular structures. *Green Chemistry* **2021**, *23*, 4451–4465.

(25) Gharagheizi, F. Quantitative structure- property relationship for prediction of the lower flammability limit of pure compounds. *Energy & fuels* **2008**, *22*, 3037–3039.