Olfactory Data Mining: Improving and Explaining the Classification of Odorant Molecules by GNN

Suman Basava[1] and Fabric Guillet[2]Angélique Villière[3]

[1]Polytech Nantes, Nantes, France
[2]LS2N, Polytech Nantes, Nantes, France
[3]GEPEA-Flaveur, Oniris, Nantes, France

Abstract

Understanding the relationship between a molecule's structure and its perceived odor, known as the Quantitative Structure-Odor Relationship (QSOR), remains a difficult problem in computational chemistry. We build on recent advances in machine learning for olfaction and explore a Graph Neural Networks (GNN)-based approach to classify molecular odors. Our method includes three main improvements: (1) Modifying and evaluating a basic GCN architecture, (2) enhancing molecular graph representations with odor-related functional group labels, and (3) incorporating semantic relationships between odor classes using the Structure of Scent (SoS) hierarchy. We perform a thorough evaluation that includes ablation studies, cross-dataset tests, and comparisons with top QSOR models. This provides insights into the importance of chemically informed graph representations and semantic label hierarchies in predicting molecular odors.

1 Introduction

Olfaction is a complex sense with many applications, including in the production of flavours and fragrances, environmental monitoring and medicine. Despite years of research, it remains extremely challenging to predict a molecule's odour based solely on its structure. Due to the complex relationship between a molecule's structure and our perception of its odour, this endeavor known as quantitative structure–odour relationship (QSOR) modelling is challenging. However, recent breakthroughs have demonstrated the feasibility of using machine learning for olfactory tasks. The "Machine Learning for Scent" paper [1] introduced GNN-based methods that outperformed traditional descriptor-driven models. "A Principal Odor Map" [2] provided a unified embedding space for various olfactory tasks, emphasizing perceptual consistency. Similarly, "Molecular Odor Prediction Based on Multi-Feature Graph Attention Networks" [3] enhanced feature extraction through attention mechanisms. However, these methods still have issues with robustness, generalization, and obtaining crucial chemical information regarding odor-related groups in molecules and they do not consider the data set with hierarchical relationship.

This work makes three key advances in this area using graph neural networks (GNNs) for molecular odor classification. First, we improve the fundamental GNN model by including more specific details about atoms, bonds, and molecules. Secondly, we annotate molecular graphs with functional groups known to influence scent, such as aldehydes, ketones, esters, and more than 30 others, based on curated sources, including OlfactionBase [4], to ensure that the selected groups are chemically and biologically relevant rather than arbitrary. Third, we use the Structure of Scent (SoS) hierarchy [5] to incorporate information on the relationships between different scents.

We evaluate our methodology using a range of experiments, including ablation studies and tests on various datasets. Our method achieved an **F1 score** of **0.384** on the validation set. These experiments help us to better understand how chemical and semantic information can improve molecular odour prediction. The code and resources for this work are available at
https://github.com/Suman1108/Molecule_Odor_Classification_with_GNN

2 Recent Works

2.1 Models and Methods

Sanchez-Lengeling et al. [1] introduced the "Machine Learning for Scent" framework, a ground-breaking deep learning method for simulating olfactory perception with graph neural networks (GNNs). Molecules were represented as graphs and trained on perceptual odor labels sourced from the GoodScents perfume materials database [6] and the Leffingwell PMP 2001 dataset [7]. They proposed two types of graph neural network (GNN) architecture: Message-Passing Neural Networks (MPNNs) and Graph Convolutional Networks (GCNs). Their experiments showed that these two architectures performed similarly. In particular, they compared these advanced neural network models with traditional machine learning approaches such as Random Forest and K-Nearest Neighbors (KNN), which use conventional fingerprint features to represent molecules. Using atom-level feature engineering within the GNNs enabled them to capture more detailed chemical information directly from the molecular structure, achieving better results and outperforming all the traditional models that used fingerprint-based features. Despite its success, the model has certain limitations. It relies on fixed molecular descriptors and moderately sized datasets and does not incorporate domain-specific chemical knowledge such as functional group annotations or hierarchical semantic relationships among odor descriptors. In this work, we adopt the Machine Learning for Scent framework as a baseline for performance benchmarking.

Lee et al. (2022) [2] trained a Message Passing Neural Network (MPNN) a type of Graph Neural Network (GNN)—to predict how molecules smell based on their structure. Each molecule was represented as a graph with detailed atomic and bond features. The model was trained on $\sim$5,000 molecules labeled with odor descriptors sourced from the GoodScents perfume materials database [6] and the Leffingwell PMP 2001 dataset [7]. The curated dataset used for training is publicly available in the replicated Git repository [17]. From this, they extracted a Principal Odor Map (POM) from the penultimate GNN layer, which successfully captured human-like odor similarities and hierarchies. The model was validated on 400 new molecules, achieving odor quality predictions that matched or exceeded the performance of human panelists. Compared to traditional chemoinformatics and random forests, the GNN showed superior generalization. The POM also outperformed other models on downstream tasks. Overall, this work offers a machine learning-based map that closely mirrors human olfactory perception. While POM offers a valuable semantic structure of odor perception, its primary focus lies in representational learning.

In the study "Molecular Odor Prediction Based on Multi-Feature Graph Attention Networks" by Xie et al [3] the authors tackle the molecular odor prediction task using a multi-level feature extraction strategy combined with a Graph Attention Network (GAT). Their feature engineering integrates fundamental atomic and bond characteristics, functional groups identified through SMARTS patterns, and global molecular fingerprints such as Morgan, MACCS, and Topological descriptors to comprehensively represent molecular features. The proposed Hierarchical Attention Graph Convolutional Network (HAGCN) utilizes multi-head attention and an attention-based aggregation mechanism to dynamically emphasize the importance of nodes, thereby effectively modeling complex molecular structures. Moreover, they incorporate an Adaptive Focal Loss function to mitigate label imbalance in multi-label classification. This approach surpasses traditional descriptor-based methods and prior GNN models, achieving higher AUROC and F1 scores, which highlights its improved accuracy and robustness in challenging Quantitative Structure-Odor Relationship (QSOR) prediction tasks.

"Mol-PECO: Directional GCN with Positional Encoding" [9], this study presents, a deep learning model developed to predict how humans perceive odors based on molecular structures. It combines the Coulomb matrix (CM), which captures atomic charges and 3D positions, with spectral positional encoding derived from graph Laplacian eigenfunctions. Unlike traditional graph convolutional networks (GCNs) that use sparse adjacency matrices and miss some 3D and global molecular information, Mol-PECO leverages the fully connected CM to better represent molecular electrostatics and structure. The model improves atom embeddings by adding spatial context through a learned spectral attention mechanism. Tested on a large dataset of 8,503 molecules with 118 odor descriptors, Mol-PECO performs better than conventional machine learning methods, showing higher AUROC and AUPRC scores. However, it still does not surpass the performance of earlier GCN and message passing neural network (MPNN) models. This work offers valuable insights into feature representation and introduces a new approach to feature engineering that could lead to more effective models in the future.

Component	**Machine Learning for Scent**	**POM**	**Mol-PECO**
Encoder: Message Passing	Concatenation message type (GCN), 4 layers with dims: [15, 20, 27, 36], SELU activation, Max graph pooling	Edge-conditioned message type (MPNN), 3 layers of dim 64,GRU update at each layer with Sum aggregation, ReLU activation	Construct graph using the CM, encodes atom positions via PE and transformer, features are updated using a GCN with CM-based edge weights and skip-connections.
Readout	Global sum pooling with softmax, 175 dim output, One per message passing layer, summed	DGL Set2Set combines final node(64) and edge(64) embeddings with internal 3-layer LSTM	sum pooling over atom embeddings
Decoder: Fully Connected NN	2 fully connected layers: [96, 63], ReLU activation, BatchNorm and dropout (0.47)	4-layers with decreasing dim from 1024 to 256 with relu, dropout of 0.12 and l1/l2 regularization	No specific info
Prediction	Multi-headed sigmoid for 138 odor tasks	Multi-task sigmoid output for 138 odor classification tasks.	Predicts 118 odor descriptors
Training	Weighted cross-entropy loss, Adam optimizer with LR decay and warm restarts, 300 training epochs	Weighted-cross entropy loss, optimized with Adam, used learning rate decay, 150 epochs	No specific info

Table 1: This table provides an overview of the model architecture used in the specific study.

2.2 Datasets and Results from Previous Studies

This subsection provides a comparative overview of prior works in terms of the datasets they used and the performance they reported. We summarize the key characteristics of each dataset, such as size, label type, and hierarchy. We also present benchmark results to highlight the progress and challenges in molecular odor prediction.

Paper	Dataset Source	# Molecules	Odor Descriptors	Data Type
Machine Learning for Scent [1]	GS, LWF	5,030	138	Flat
POM [2]	GS, LWF	~5,000	138	Flat
Multi-Feature GAT [3]	GS, LWF	5,788	154	Flat
Mol-PECO [9]	PYR	8,503	118	Flat

Table 2: Summary of datasets used in prior studies. Dataset abbreviations — GS: GoodScents [6], LWF: Leffingwell [7], PYR: Pyrfume [18]

Paper	AUROC	Precision	Recall	F1-score
Machine Learning for Scent [1]	0.894	0.379	0.387	0.362
POM [2]	0.854	0.578	0.557	0.555
Multi-Feature GAT [3]	0.9294	-	-	0.4632
Mol-PECO [9]	0.813	0.104	0.819	0.185

Table 3: Reported performance of prior studies on odor prediction tasks. Metrics are reported as published in respective papers. POM does not report these metrics directly except AUROC, as its focus is representational learning and the other metrics are computed from their GitHub repository [17].

3 Proposed Work

3.1 Data Collection

We collected the initial dataset from the GoodScents perfume materials database [6], which contains curated mappings between chemical compounds and their associated odor descriptors. Each molecule was initially linked to a set of perceptual odor terms. To enhance semantic consistency and reduce ambiguity, these terms were mapped into a standardized olfactory taxonomy known as the SketchOscent odor space [5]. To further enrich the dataset, we performed *saturation* of the odor labels. Saturation here refers to the process of expanding and hierarchically enriching the original odor labels using the structured knowledge graph provided by the SketchOscent ontology. This knowledge graph captures relationships such as "is-a" and "part-of" between odor descriptors (e.g., `citrus` → `fruity` → `pleasant`), allowing more comprehensive odor recognition for each molecule.

After saturation, the dataset consisted of 3,687 molecules labeled with 436 distinct odor descriptors. Each molecule was identified by its CAS (Chemical Abstracts Service) number, which served as a unique key to retrieve corresponding SMILES (Simplified Molecular Input Line Entry System) representations. SMILES strings were extracted using two sources: PubChemPy [10] and the CACTUS [11] chemical resolver. These structures were then validated and standardized. For compatibility with the baseline model, the dataset was taken from the OpenPOM repository [17]. This dataset replicates the one used in the Principal Odour Map (POM).

The dataset comprises 138 odour descriptor annotations for over 5,000 molecules. The original odour descriptors were then mapped to their respective words in the Sketch-Oscent (SoS) taxonomy to provide hierarchical, semantically enhanced labels. For more detail, refer to Appendix A.

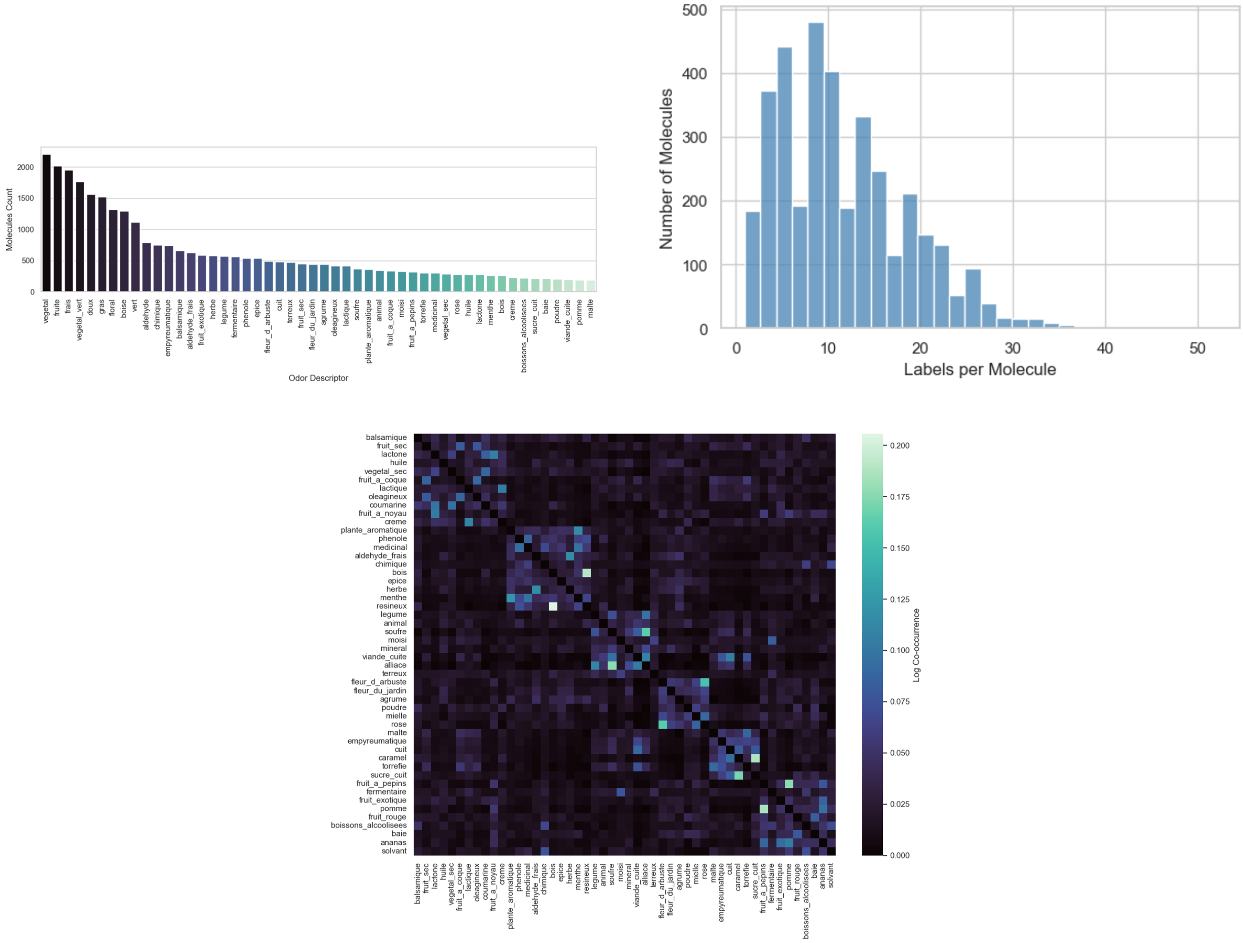

Figure 1: Dataset overview. A: Distribution of odor descriptor frequencies. B: Distribution of label density. C: This heatmap visualizes the co-occurrence matrix of the top 50 odor descriptors, reordered using spectral clustering . The rows and columns represent odor descriptors, with darker shades indicating lower co-occurrence and lighter shades (green) indicating higher co-occurrence. Spectral clustering groups similar descriptors together, revealing distinct clusters such as fruity, plant-based, and exotic scents.

3.2 Featurization

We employ a multi-scale featurization strategy to encode molecules as graphs suitable for graph neural networks. Each molecule is represented as a graph $G = (V, E)$, where atoms are nodes $v_i \in V$, and bonds are edges $e_{ij} \in E$. For each node and edge, we extract chemically meaningful features that capture local atomic structure, bonding context, and global molecular descriptors [12], forming a comprehensive representation suitable for odor prediction. In total, we use 66 features to perform this featurization, encompassing atomic properties, bond types, and global molecular characteristics (including functional groups).

Atomic Features: Each atom A_i is encoded using a feature vector $\mathbf{F}(A_i) \in \mathbb{R}^{d_n}$, where $d_n = 15$. These features are derived directly from RDKit [13] and include both topological and electronic descriptors. Basic integer-valued attributes include atomic number (Z_i), degree (D_i), valence (V_i), formal charge (Q_i), number of hydrogen atoms (H_i), and number of radical electrons (R_i). Additionally, binary flags are used to indicate whether the atom is aromatic (A_i) and whether it resides within any ring structure (Rg_i).

Geometric and stereochemical descriptors are also encoded. The size of the smallest ring (S_i) an atom belongs to is included as an integer value, providing structural context that can influence molecular flexibility and conformational strain. Chirality (C_i) and hybridization (Hy_i) are categorical properties, encoded as integers. Chirality includes values such as `CHI_TETRAHEDRAL_CW`, `CHI_TETRAHEDRAL_CCW`, and others, while hybridization includes `SP`, `SP2`, `SP3`, and other states. Bond-type context is encoded at the atomic level using a multi-hot vector representing the types of bonds the atom is connected to. This includes binary flags for SINGLE, DOUBLE, TRIPLE, and AROMATIC bond types:

$$\mathbf{B}_i = [b_{\text{single}}, b_{\text{double}}, b_{\text{triple}}, b_{\text{aromatic}}] \in \{0, 1\}^4$$

The final atomic feature vector is given by:

$$\mathbf{F}(A_i) = [Z_i, D_i, Q_i, H_i, R_i, V_i, A_i, Rg_i, S_i, C_i, Hy_i, \mathbf{B}_i]$$

where each symbol refers to the attributes described above.

Molecular-Level Features: For each molecule, we compute a set of global physicochemical descriptors $\mathbf{f}_m \in \mathbb{R}^{51}$. These features summarize the overall shape, reactivity, and bioavailability characteristics of the molecule and are known to correlate with odor intensity and receptor activation. The features include molecular weight (MW), logP, topological polar surface area (TPSA), and the total formal charge (Q_{total}). Topological features such as the number of rotatable bonds (N_{rot}), number of rings (N_{rings}), number of hydrogen bond donors (N_{donors}) and acceptors ($N_{\text{acceptors}}$), and heavy atom count (N_{heavy}) are included to capture molecular flexibility and interaction potential. Additionally, the longest linear carbon chain (LCC) is computed using depth-first search, which serves as a proxy for chain length and molecular geometry. Molecular complexity and fraction of sp^3 hybridized carbon atoms are also incorporated to reflect the saturation and branching of the structure.

Functional Group Features: To provide chemically rich context, we integrate SMARTS-based substructure matching to detect and count occurrences of 40 functional groups known to influence odor perception. For a molecule M and a predefined set of functional group patterns $G = \{G_1, G_2, \ldots, G_n\}$, the functional group feature vector is constructed as:

$$f_i = \text{count of occurrences of functional group } G_i \text{ in } M, \tag{1}$$
$$\mathbf{f}_{\text{func}}(M) = [f_1, f_2, \ldots, f_n] \in \mathbb{N}^n,$$

Thus the above equation represents the functional group feature vector with each component representing the number of times the corresponding functional group G_i appears in molecule M. These counts are concatenated with the global molecular descriptors to form the complete molecular feature vector:

$$\mathbf{f}_{\text{mol}} = [\text{MW}, \log P, \text{TPSA}, N_{\text{rings}}, N_{\text{rot}}, N_{\text{donors}}, N_{\text{acceptors}}, N_{\text{heavy}}, Q_{\text{total}}, C_{\text{sp3}}, LCC, \mathbf{f}_{\text{func}}]$$

where MW is molecular weight, TPSA is topological polar surface area, and other symbols correspond to standard molecular descriptors as defined earlier. For additional details about the features used, please refer to Appendix B

3.3 Model Architecture and Training Pipeline

The training pipeline was designed with a focus on enhancing model performance for molecular odor prediction by systematically replacing baseline choices with more effective alternatives. One of the major changes was the replacement of the standard weighted cross-entropy loss function [14] with the focal loss. Focal loss [15] is particularly suitable for multi-label classification problems with severe label imbalance, as is the case in odor datasets. In fact, Focal Loss is a kind of reshaped cross entropy loss that the weighs of well classified examples are reduced. Formally, Focal Loss is defined as:

$$\mathcal{F}L(p) = -\Big(y(1-p)^{\gamma}\log p \ + \ (1-y)p^{\gamma}\log(1-p)\Big)$$

where $y \in \{0, 1\}$ specifies the ground-truth class, $p \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$, and γ is a tunable focusing parameter. When $\gamma = 0$, the focal loss is equivalent to the cross-entropy loss, and as γ increases, the effect of the modulating factor also increases. The focal loss focuses training on a sparse set of hard examples and prevents the large number of easy negatives from overwhelming the classifier during training. Another change was the use of saturated and enriched data sampling instead of raw or randomly selected examples. This ensured that molecules associated with frequent odor descriptors were better represented during training. Stratified sampling using `MultilabelStratifiedKFold` [16], as originally used in the baseline method [1], was replaced with a customized stratified splitting strategy inspired by DeepChem's `RandomStratifiedSplit` implementation. This alternative approach offered improved control over stratification and yielded more stable and consistent label distribution across the folds, thereby enhancing validation performance. Hence, adopting DeepChem's stratification logic during hyperparameter tuning led to a notable improvement in F1 score. This suggests that effective data balancing is one of the most critical factors in multi-label classification problems.

The final model architecture introduces several important changes compared to the baseline. Instead of using a deeper GCN with many layers, it adopts a simpler design with two GINConv layers of sizes 20 and 155. The first layer processes the 15-dimensional input node features, followed by batch normalization, SELU activation, and dropout (0.2), then hierarchical SAGPooling. SAGPooling [19] is a pooling method that selects important nodes based on self-attention scores, helping to reduce graph size while preserving key information. After each GIN layer, global add pooling is applied, and the outputs are concatenated into a 175-dimensional vector. This is combined with a 51-dimensional molecular feature vector, forming a 226-dimensional representation. Finally, this combined vector is passed through a two-layer MLP with hidden sizes [100, 70] to predict the 138 odor classification tasks.

The decoder, implemented as a multi-layer perceptron (MLP), uses a dropout rate of 0.2 to regularize the training process. Optimization was performed using Adam with L2 regularization to mitigate overfitting caused by the increased input dimensionality. The readout layer aggregated graph embeddings using global add pooling, and no softmax was applied to the final logits to maintain compatibility with the multi-label focal loss.

The training pipeline utilizes random stratified cross-validation to split the data while preserving label distributions. For each fold, the model is trained using the Adam optimizer with weight decay, minimizing a focal loss to handle class imbalance. Training proceeds in mini-batch of size 32, computing predictions and updating model weights accordingly. After training, the model is evaluated on the validation set using metrics such as accuracy, micro F1-score, and AUROC. Metrics are recorded for each fold, and final statistics are aggregated across all folds to assess overall

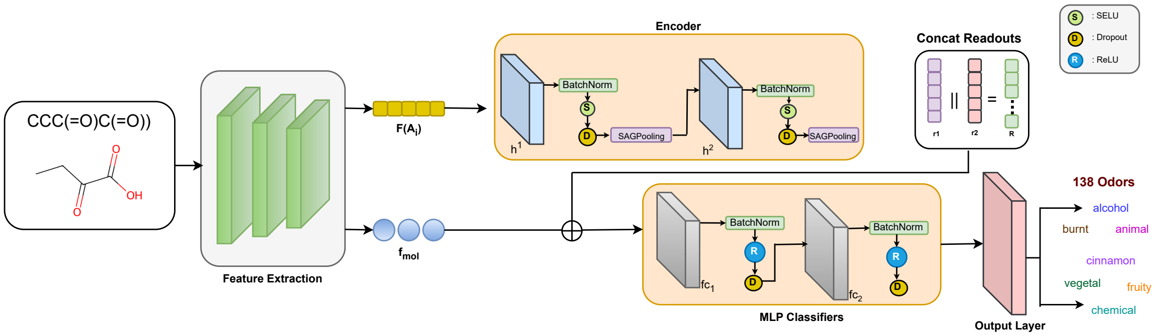

Figure 2: Overview of the proposed model architecture for molecular odor prediction. The input consists of a molecule's SMILES string, from which node-level features $F(A_i)$ and global molecular features f_{mol} are extracted. These are processed through two graph convolutional layers, producing intermediate representations h_1 and h_2, each followed by SELU activation and global add pooling. The resulting graph-level embeddings are concatenated with f_{mol} to form a unified representation, which is passed to a two-layer MLP classifier. SELU and ReLU activations are color-coded in the figure. The final output predicts 138 odor descriptors. *Figure created using diagrams.net.*

performance.

Comparison with Prior Architectures: Compared to the models discussed in the Related Work section, the proposed approach explores a simpler and more interpretable architecture, emphasizing conceptual clarity over complexity. Unlike the baseline, which employs a deeper four-layer GCN with increasing dimensions and max pooling, this work uses a two-layer GCN encoder with SELU activation and global add pooling to limit over-smoothing and maintain interpretability. Additionally, MLFS applies a softmax function at the final readout layer, which is suboptimal in multi-label classification tasks—softmax forces the output probabilities to sum to one, thereby treating the task as mutually exclusive, which suppresses the prediction of multiple co-occurring labels. To address this, we omit softmax and use independent sigmoid activations instead, allowing the model to assign probabilities independently for each of the 138 odor descriptors. The DeepChem framework's implementation of the POM model features a very intricate, tiered design that aims to provide a single odour embedding space. A recurrent Set2Set readout and edge-conditioned MPNNs with GRU-based iterative updates are used to integrate node and edge embeddings via LSTM processes. The interpretability and simplicity of this complex architecture are sacrificed in favour of rich representational learning. In contrast to the suggested lightweight, completely feedforward PyTorch-based method, DeepChem's abstractions enable such complex, TensorFlow-based models. By eschewing recurring and embedding-heavy elements, the new paradigm puts clarity and simplicity of analysis ahead of architectural complexity. Unlike Mol-PECO, which constructs fully connected graphs using Coulomb matrices and augments GCNs with positional encodings and Transformer-based updates, our method emphasizes chemically meaningful graph structures and handcrafted molecular features—avoiding the overhead of learned positional priors while retaining high predictive performance. Crucially, to handle the inherent class imbalance in odor datasets, we replace the weighted cross-entropy loss used in baseline and POM with focal loss, which significantly contributed to the performanc on minority labels by focusing learning on hard, misclassified examples. This combination of architectural restraint, interpretability, and targeted enhancements makes our model both efficient and robust for multi-label molecular odor prediction.

4 Experiments

4.1 Model Comparison

Our proposed model builds on the baseline architecture from "Machine Learning for Scent" [1] by incorporating modifications in data processing, model architecture, and training strategy. The final model employs a 2-layer GIN encoder which is trained on stratified, saturated data representations and includes both molecular descriptors and SMARTS-based functional group counts as global features. We use a focal loss function to handle class imbalance and omit softmax activation at the readout layer. To understand the role of each component, ablation experiments systematically remove individual parameters while keeping others fixed.

Understanding the Influence of Key Model Components

The ablation studies presented in Table 4 serve as a form of hyperparameter tuning, systematically exploring the effect of individual architectural and training components on the model's performance. By selectively removing or modifying key elements such as normalization layers, regularization methods, the focal loss function, or specific input features like global descriptors, functional groups, and the SoS hierarchy we effectively tune the model's configuration. This process allows us to identify which components contribute most significantly to accuracy and robustness. Through these controlled experiments, we observe how different parameter choices and feature sets affect the model's learning dynamics and generalization ability. The insights gained from this tuning approach guide us in optimizing the overall architecture and training procedure. Detailed analysis of each component's influence is further discussed in the discussion section.

#	Model Variant	AUROC [CI]	Precision [Min, Max]	Recall [Min, Max]	F1 Score [Min, Max]
1	Baseline GCN (BP) [1]	0.894 [0.888, 0.902]	0.379 [0.351–0.398]	0.390 [0.365–0.412]	0.360 [0.337–0.372]
2	OpenPOM_Git [17]	0.854	0.578	0.557	0.555
3	**GIN_Model**	**0.879 [0.869, 0.889]**	**0.376 [0.370, 0.381]**	**0.512 [0.494, 0.531]**	**0.384 [0.386, 0.394]**
4	Without Focal Loss	0.864 [0.863, 0.865]	0.648 [0.634, 0.662]	0.253 [0.243, 0.262]	0.277 [0.269, 0.283]
5	Without Normalizers	0.840 [0.836, 0.845]	0.496 [0.484, 0.508]	0.364 [0.358, 0.371]	0.271 [0.264, 0.278]
6	Without Regularizer	0.859 [0.857, 0.862]	0.372 [0.359, 0.385]	0.459 [0.453, 0.466]	0.332 [0.325, 0.338]
7	With Uniform Sampling	0.849 [0.844, 0.853]	0.383 [0.373, 0.432]	0.431 [0.421, 0.432]	0.311 [0.308, 0.314]
8	Without Global Features	0.844 [0.836, 0.853]	0.508 [0.454, 0.564]	0.274 [0.253, 0.295]	0.208 [0.190, 0.225]
9	Without Softmax at Readout	0.870 [0.869, 0.872]	0.352 [0.352, 0.356]	0.506 [0.499, 0.513]	0.370 [0.368, 0.373]
10	With Raw Dataset	0.842 [0.709, 0.868]	0.478 [0.401, 0.511]	0.291 [0.274, 0.375]	0.224 [0.219, 0.290]
11	Without Graphpooling	0.869 [0.868, 0.871]	0.343 [0.335, 0.353]	0.526 [0.523, 0.530]	0.376 [0.375, 0.376]
12	With GCN Model	0.863 [0.859, 0.866]	0.371 [0.353, 0.388]	0.483 [0.483, 0.483]	0.353 [0.348, 0.359]
13	With 4 Layers	0.860 [0.857, 0.863]	0.371 [0.357, 0.384]	0.463 [0.462, 0.464]	0.345 [0.342, 0.349]

Table 4: **Validation results for odor descriptors across models and ablation studies.** The table presents the mean and 95% confidence intervals [lower, upper] for AUROC, along with the ranges [min, max] for Precision, Recall, and F1 Score. All metrics are computed as weighted averages across 138 odor descriptors. The first two rows present comparison models; the proposed model is in **bold**, and the other rows show modifications made to the proposed model.

Table of Per-Descriptor Results

AUROC and F1 Score performance results by descriptor for proposed GIN model.

No.	Descriptor	AUROC	F1 Score
1	Onion	0.9720	0.5276
2	Garlic	0.9716	0.4947
3	Alliaceous	0.9682	0.5851
4	Sulphurous	0.9666	0.6968
5	Odorless	0.9569	0.7010
6	Sulphur	0.9483	0.6175
7	Vanilla	0.9442	0.4843
8	Alcohol	0.9395	0.5430
9	Banana	0.9379	0.4153
10	Smoked	0.9338	0.3151
11	Smoky	0.9323	0.3145
12	Ethereal	0.9302	0.5350
13	Cognac	0.9294	0.3563
14	Coffee	0.9292	0.2927
15	Musk	0.9279	0.5502
16	Musky	0.9269	0.5591
17	Meaty	0.9236	0.4790
18	Solvent	0.9230	0.5643
19	Camphor	0.9186	0.3599
20	Pine	0.9158	0.2586
21	Wood	0.9153	0.3539
22	Pear	0.9119	0.3412
23	Lemon	0.9091	0.2367
24	Lemony	0.9073	0.3302
25	Resinous	0.9067	0.3025
26	Potato	0.9066	0.1115
27	Roasted	0.9032	0.4970
28	Lily	0.8980	0.1527
29	Vegetable	0.8976	0.5552
30	Caramel	0.8972	0.4557
31	Cooked sugar	0.8970	0.4526
32	Cruciferous	0.8968	0.3072
33	Winey	0.8961	0.3448
34	Fermented drink	0.8957	0.3423
35	Savory	0.8951	0.2798
36	Apple	0.8943	0.4098
37	Cinnamon	0.8938	0.2460
38	Coconut	0.8910	0.3842
39	Muguet	0.8886	0.1485
40	Burnt	0.8875	0.3304
41	Waxy	0.8849	0.4948
42	Alcoholic beverages	0.8848	0.4345
43	Almond	0.8833	0.2321
44	Empyreumatic	0.8831	0.5503
45	Bitter almond	0.8822	0.2252
46	Fruit with pip	0.8810	0.4520
47	Amber	0.8775	0.3430
48	Butter	0.8769	0.3112
49	Cheese	0.8722	0.3551
50	Pungent	0.8715	0.2764
51	Chocolate	0.8714	0.1517
52	Coumarin	0.8696	0.3568
53	Pineapple	0.8685	0.3462
54	Tropical fruit	0.8679	0.4248
55	Violet	0.8674	0.1723
56	Nutty	0.8611	0.4650
57	Cooked	0.8611	0.3869
58	Barrel aged alcohol	0.8601	0.3053
59	Medicinal	0.8593	0.4116
60	Malt	0.8591	0.3069
61	Hyacinth	0.8576	0.1440
62	Mint	0.8572	0.3978
63	Cocoa	0.8564	0.3092
64	Oleaginous	0.8556	0.4856
65	Cherry	0.8553	0.2084
66	Dried fruit	0.8545	0.4857
67	Melon	0.8528	0.3038
68	Grapefruit	0.8523	0.2339
69	Phenolic	0.8520	0.4680
70	Peponide	0.8519	0.3023
71	Peach	0.8463	0.2958
72	Aromatic	0.0199	0.8410
73	Fishy	0.2838	0.8405
74	Sharp	0.1172	0.8384
75	Anisic	0.2064	0.8383
76	Cooling	0.2435	0.8365
77	Citrus	0.4148	0.8351
78	Sour	0.2772	0.8351
79	Tobacco	0.1264	0.8345
80	Apricot	0.0209	0.8291
81	Rummy	0.1656	0.8286
82	Jasmin	0.2331	0.8286
83	Clean	0.0110	0.8264
84	Aromatic Plant	0.3310	0.8263
85	Stone Fruit	0.2300	0.8214
86	Rose	0.3093	0.8185
87	Orris	0.1312	0.8140
88	Aldehydic	0.3452	0.8121

No.	Descriptor	AUROC	F1 Score
89	Grape	0.2479	0.8104
90	Animal	0.3548	0.8082
91	Chemical	0.4551	0.8078
92	Orange	0.1162	0.8073
93	Garden Flower	0.2895	0.8040
94	Ripe	0.1544	0.8038
95	Bush Flower	0.3127	0.8034
96	Lactone	0.3502	0.8031
97	Plum	0.0071	0.8030
98	Fresh Aldehydic	0.2168	0.8026
99	Powdery	0.1439	0.8021
100	Tropical	0.3724	0.8003
101	Hay	0.1288	0.7982
102	Floral	0.5663	0.7948
103	Honeyed	0.2843	0.7946
104	Oily	0.3627	0.7940
105	Honey	0.2862	0.7938
106	Fatty	0.5280	0.7930
107	Fruity	0.7247	0.7912
108	Mushroom	0.1078	0.7883
109	Balsamic	0.7824	0.4088
110	Geranium	0.7818	0.0722
111	Fresh	0.7817	0.6647
112	Spicy	0.7807	0.3455
113	Dairy	0.7769	0.3627
114	Dry vegetal	0.7746	0.2882
115	Fermented	0.7742	0.4135
116	Woody	0.7726	0.5213
117	Green	0.7718	0.5209
118	Red berry	0.7593	0.1013
119	Milky	0.7559	0.1214
120	Bitter	0.7552	0.0411
121	Grass.1	0.7522	0.3818
122	Green vegetal	0.7522	0.5884
123	Cream	0.7425	0.2056
124	Grass	0.7373	0.0218
125	Vegetal	0.7358	0.6834
126	Leafy	0.7343	0.0715
127	Sweet	0.7331	0.6084
128	Berry	0.7320	0.0799
129	Dry	0.7277	0.1446
130	Warm	0.7240	0.0009
131	Metal	0.7233	0.0133
132	Mineral	0.7209	0.0109
133	Metallic	0.7181	0.0126
134	Earthy	0.7141	0.2582
135	Natural	0.7133	0.0000
136	Cortex	0.7128	0.0000
137	Tea	0.7041	0.0000
138	Musty	0.6940	0.1605

This table presents a performance evaluation of various odor descriptors using two key metrics: the Area Under the Receiver Operating Characteristic curve (AUROC) and the F1 Score. AUROC measures the model's ability to distinguish positive from negative samples, while the F1 Score reflects the balance between precision and recall. Some descriptors have an F1 Score of zero despite decent AUROC values because the model fails to confidently predict any positive cases for those labels. This usually happens when the descriptors are extremely rare or have very few examples, causing the model to miss them during prediction.

5 Discussion

Influence of Individual Model Parameters

To understand the contribution of individual components in the proposed GIN Model, we conducted extensive ablation experiments and analyzed their effect on validation metrics. Table 4 summarizes the results. Below, we discuss the role and impact of each parameter:

- **Focal Loss**: Removing focal loss (Row 4) resulted in a sharp drop in F1 Score from 0.384 to 0.277, despite an increase in precision to 0.648. This confirms that focal loss is essential for handling class imbalance by emphasizing difficult (rare) odor descriptors and avoiding domination by frequent classes.

- **Normalization**: Excluding normalization layers (Row 5) led to a drop in AUROC from 0.879 to 0.840 and F1 Score from 0.384 to 0.271, suggesting that normalization promotes smoother training dynamics and stabilizes feature learning across the network layers.

- **Regularization (L1/L2)**: Surprisingly, removing the regularizer (Row 6) slightly improved recall (from 0.512 to 0.459), though F1 Score dropped modestly (to 0.332). This suggests that, despite the risk of overfitting, the model may be slightly under regularized and could benefit from greater flexibility.

- **Global Features (Descriptors + SMARTS)**: Eliminating chemically rich global features (Row 8) led to a significant performance drop across all metrics, with F1 falling to 0.208 and AUROC to 0.844. This highlights the importance of complementary chemical descriptors in enhancing graph-based molecular representations.

- **SoS Odor Hierarchy**: When odor labels were used in a flat structure (reflected in degraded precision/recall in Rows 8), the model's ability to differentiate between related classes suffered. Hierarchical structure (as used in the full model) helps in capturing semantic similarities between descriptors.

- **Sampling Strategy**: Using uniform sampling rather than stratified sampling (Row 7) improved recall (0.431) but reduced F1 (0.311) and AUROC (0.849), confirming that stratified sampling leads to better trade-offs between minority and majority classes and avoids skewed performance.

- **Softmax Activation at Readout**: Including a softmax layer before the final MLP readout (Row 9) slightly decreased precision and F1 Score, indicating that in a multi-label setting, independent sigmoid outputs (without softmax) are more effective for making non-exclusive predictions. This was considered because the baseline paper mentions the use of a softmax layer at the readout stage.

- **Data Saturation (Raw vs Enriched)**: Using the raw dataset (Row 10) without saturation substantially lowered F1 Score to 0.224 and AUROC to 0.842, confirming that enriched descriptors (e.g., Apple → Fruity) play a key role in improving generalizability and semantic resolution.

- **Graph Pooling and Model Depth**: Removing graph pooling (Row 11) or changing architecture depth (Row 13) both led to slight performance drops, suggesting that architectural design choices (e.g., layers, pooling) contribute marginal but consistent gains to the overall model efficacy.

6 Conclusion

In this study, we propose a GIN-based graph neural network framework for predicting multi-label odour descriptors. This framework integrates structural graph representations with chemically enriched global features. To address class imbalance, the model incorporates focal loss and leverages hierarchical saturation of odour descriptors to capture semantic relationships between related odour classes more effectively. Our ablation studies highlight the importance of specific design choices.

In particular, augmenting global molecular descriptors with functional group-based features provided complementary signals beyond the graph structure. Hierarchical saturation increased the sensitivity of the model to fine-grained odour categories, and focal loss significantly improved recall by focusing on samples that were difficult to classify. Overall, the proposed framework achieves strong, consistent performance across key metrics, demonstrating the advantages of combining molecular graph learning with chemically informed priors. This work establishes a robust foundation for future advances in interpretable, generalisable and scalable odour prediction systems, making a meaningful contribution to data-driven olfactory modelling.

References

[1] Benjamin Sanchez-Lengeling, Jennifer N Wei, Brian K Lee, Richard C Gerkin, Alán Aspuru-Guzik, and Alexander B Wiltschko, "Machine learning for scent: Learning generalizable perceptual representations of small molecules," *arXiv preprint arXiv:1910.10685, 2019*.

[2] A Principal Odor Map Unifies Diverse Tasks in Human Olfactory Perception. Brian K. Lee, Emily J. Mayhew, Benjamin Sanchez-Lengeling, Jennifer N. Wei, Wesley W. Qian, Kelsie A. Little, Matthew Andres, Britney B. Nguyen, Theresa Moloy, Jacob Yasonik, Jane K. Parker, Richard C. Gerkin, Joel D. Mainland, Alexander B. Wiltschko *Nature*, 601(7891), 177–182.

[3] Zhou, X., Xu, C., Wu, Y., Zheng, Y., & Zhao, X. (2022). Molecular odor prediction based on multi-feature graph attention networks. *Frontiers in Chemistry*, 10, 867529.

[4] Sharma A, Saha BK, Kumar R, Varadwaj PK. OlfactionBase: a repository to explore odors, odorants, olfactory receptors and odorant-receptor interactions. Nucleic Acids Res. 2022 Jan 7;50(D1):D678-D686. doi: 10.1093/nar/gkab763. OlfactionBase. (n.d.). Chemicals. IIIT Allahabad. Retrieved July 22, 2025, from https://olfab.iiita.ac.in/olfactionbase/chemicals

[5] Angélique Villière, Catherine Fillonneau, Carole Prost, Fabrice Guillet. SketchOscent: towards aknowledge-based model and interactive visualisation of the odour space. Proceedings of the 16thWeurman Flavour Research Symposium, 2022, 10.5281/zenodo.5948778. hal-03556965 *https://oniris-polytech.univ-nantes.io/sketchoscent*.

[6] The good scents company - flavor, fragrance, food and cosmetics ingredients information. http://www.thegoodscentscompany.com/. Accessed: 2019-9-4.

[7] John C Leffingwell. Leffingwell & associates, 2005.

[8] Structure of Scents (SoS) Dataset, https://oniris-polytech.univ-nantes.io/sketchoscent, 2023.

[9] Zhang, M., Hiki, Y., Funahashi, A. et al. A deep position-encoding model for predicting olfactory perception from molecular structures and electrostatics. npj Syst Biol Appl 10, 76 (2024). https://doi.org/10.1038/s41540-024-00401-0

[10] NIH - National Library of Medicine *https://pubchem.ncbi.nlm.nih.gov*

[11] NCI/CADD Chemical Identifier Resolver *https://cactus.nci.nih.gov/chemical/structure*.

[12] PyTorch Geometric. *torch_geometric.utils.smiles Documentation*. Available at: https://pytorch-geometric.readthedocs.io/en/2.4.0/_modules/torch_geometric/utils/smiles.html.

[13] RDKit Project. *RDKit C++ API: RDKit::Atom Class Reference*. Available at: https://www.rdkit.org/docs/cppapi/classRDKit_1_1Atom.html.

[14] PyTorch. torch.nn.CrossEntropyLoss — PyTorch Documentation. Available at: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[15] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. *Focal Loss for Dense Object Detection*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. Available at: https://doi.org/10.1109/ICCV.2017.324.

[16] Sechidis, K., Tsoumakas, G., and Vlahavas, I. *On the Stratification of Multi-label Data*. In: Gunopulos, D., Hofmann, T., Malerba, D., and Vazirgiannis, M. (eds) *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2011*. Lecture Notes in Computer Science, vol 6913. Springer, Berlin, Heidelberg, 2011, pp. 145–158. Available at: https://doi.org/10.1007/978-3-642-23808-6_10.

[17] OpenPOM Dataset. *An Open-source Dataset for Molecular Odor Prediction*. GitHub repository, 2024. Available at: https://github.com/ARY2260/openpom.

[18] Pyrfume Contributors. Pyrfume Data Repository. Available at: https://github.com/pyrfume/pyrfume-data

[19] Lee J, Lee I, Kang J (2019) Self-attention graph pooling. In: Proceedings of the 36th international conference on machine learning. pp 3734–3743

Supplementary Detailed Information

A Dataset Collection and Preprocessing

The initial dataset was obtained from the **GoodScent** company, where each molecule is annotated with odor descriptors mapped to the **Scents of Scent (SoS) ontology**. The raw dataset consisted of **3,841 molecules**, each identified by a unique *CAS number* and annotated with **394 odor descriptors**. To enable graph-based molecular representation learning, the *SMILES representations* of the molecules were generated. A two-step retrieval process was adopted to maximize SMILES coverage. Initially, the **CACTUS chemical identifier resolver API** was queried using the CAS numbers, which successfully returned SMILES strings for over **2,000 molecules**. For the remaining CAS numbers where CACTUS did not yield results, the **PubChemPy API** was employed as a fallback. This combined approach ensured near-complete coverage of SMILES generation. However, **141 CAS numbers** could not be resolved to any valid SMILES string in available open-source APIs and were therefore excluded from further analysis. The resulting dataset comprised **3,700 molecules**, each with a valid SMILES string and annotated with the original **394 odor descriptors**.

To incorporate hierarchical label information, the dataset was saturated by propagating parent–child relationships present in the **SoS ontology**. For example, molecules annotated with the parent label *fruit* also received its child labels such as *apple* and *orange*. This process was made possible thanks to the **Sketch-O-Scent ontology knowledge graph**, which encodes hierarchical relationships among odor descriptors in a `.ttl` (Turtle) file. Parsing and traversing this ontology allowed systematic expansion of molecular annotations to reflect the full hierarchy. And the larger molecules which has molecular weight > 600 g/mole are removed as they were not relevant to be considered. After ontology-based saturation, the dataset retained **3,687 molecules**, now annotated with an expanded set of **436 odor descriptors**. As expected, the label imbalance ratio increased due to the introduction of rare, fine-grained labels. To ensure comparability with the baseline work [1] and the **OpenPOM** benchmark dataset, only the **top 138 most frequent odor descriptors** were retained for model training and evaluation. This reduced label set offers a balance between diversity and class distribution.

Dataset	#Molecules	#Odor Descriptors	Std. Dev.	Min Count	Max Count	Mean
Raw SoS Data	3,841	394	129.94	1	1262	51.46
Data with SMILES	3,700	394	126.15	1	1220	49.99
Saturated Data	3,687	436	261.82	1	2210	97.97
Top138 Saturated Data	3,687	138	407.42	43	2208	284.05
OpenPOM Data	4,983	138	264.02	31	1902	176.17

Table 5: Dataset statistics at each processing stage: the raw data, the SMILES-resolved subset, the saturated ontology-enriched dataset, the final top-138 descriptors subset, and the OpenPOM dataset. The table shows molecule counts, number of odor descriptors, and label distribution statistics (standard deviation, minimum and maximum counts, and mean).

In addition to the GoodScent-derived dataset, the dataset from the **OpenPOM** GitHub repository was also utilized. For compatibility with the ontology-based saturation process, its label descriptors were mapped to the SoS ontology. This OpenPOM-based dataset was primarily employed in ablation experiments to evaluate individual components of the proposed approach.

Addressing imbalance of the dataset: We analyzed the class distribution in our multilabel dataset and found a strong imbalance between odor categories. The most frequent label, **fruity**, appears in **53.40%** of the samples, while the least frequent label, **muguet**, appears in only **1.12%**. The median frequency across all labels is just **3.15%**, and the variation (standard deviation) is **9.25%**, showing that some odors are much more common than others. This means the most common odor is about **47 times** more frequent than the rarest one.

This imbalance can make training difficult because the model may focus too much on the common odors and ignore the rare ones. As a result, it might achieve good overall scores but fail to correctly predict rare odors. To address this problem, we used **multilabel stratified splits** to ensure even rare odors are present in each fold of the data, and we applied **focal loss** to give more weight to rare and hard-to-predict examples during training.

B Feature Engineering Strategy

The feature selection focused on fundamental atomic properties of the molecules, including atomic number, degree, hybridization, aromaticity, and more. Given the complexity of handling explicit edge features in graph neural networks (GNNs), the edge features were integrated into the node features. This integration simplifies a representation and facilitates more efficient transformations within the GNN architecture.

Feature Category	**Features**	**Value Range**	**Encoding Type**
Node Features	1. Atomic Number	range(0, 35)	Integer
	2. Degree	range(0, 4)	Integer
	3. Formal Charge	range(-2, 2)	Integer
	4. Number of H	range(0, 5)	Integer
	5. Number of radical electrons	range(0, 1)	Integer
	6. Valence	range(0, 6)	Integer
	7. Is aromatic	[0, 1]	Boolean
	8. Is in ring	[0, 1]	Boolean
	9. Smallest Ring	range(0, 15)	Integer
	10. Chirality Chi_Unspecified: 0 Chi_Tetrahedral_cw: 1 Chi_Tetrahedral_ccw: 2 Chi_Other: 3 Chi_Tetrahedral: 4 Chi_Allene: 5 Chi_Squareplanar: 6 Chi_Trigonalbipyramidal: 7 Chi_Octahedral: 8	[0, 8]	Enum-Integer
	11. Hybridization Unspecified: 0 S: 1 SP: 2 SP2: 3 SP3: 4 SP3D: 5 SP3D2: 6 Other: 7	[0, 7]	Enum-Integer
	Bond type connected 12. Single 13. Double 14. Triple 15. Aromatic	[0, 1]	Boolean

Table 6: Node-level features used for model input, their value ranges, and encoding types.

Feature Category	**Features**	**Value Range**	**Encoding Type**
Molecular Features	1. Molecular Weight	[0]	Float
	2. LogP	[0]	Float
	3. TPSA	[0]	Float
	4. Number of rings	range(0, 38)	Integer
	5. Number of rotable bonds	range(0, 149)	Integer
	6. Number of H donors	range(0, 116)	Integer
	7. Number of H acceptors	range(0, 191)	Integer
	8. Heavy atom count	range(0, 419)	Integer
	9. Formal charge	range(-2, 2)	Integer
	10. Complexity	[0]	Float
	11. Longest carbon chain	[0]	Integer
Funtional Groups Feature	1. Acid	range(0, 3)	Integer
	2. Acetamide	range(0, 7)	Integer
	3. Alcohols	range(0, 24)	Integer
	4. Acetyl	range(0, 2)	Integer
	5. Aldehydes	range(0, 2)	Integer
	6. Alkanes	range(0, 57)	Integer
	7. Amide	range(0, 7)	Integer
	8. Amine	range(0, 4)	Integer
	9. Bicyclic	range(0, 20)	Integer
	10. Cyclic	range(0, 56)	Integer
	11. Carbonyl	range(0, 8)	Integer
	12. Esters	range(0, 8)	Integer
	13. CarboxylicAcid	range(0, 3)	Integer
	14. Ethers	range(0, 16)	Integer
	15. Cyclopropyl	range(0, 2)	Integer
	16. Furan	range(0, 3)	Integer
	17. Hydrocarbons	range(0, 40)	Integer
	18. Ethoxy	range(0, 88)	Integer
	19. Imino	range(0, 3)	Integer
	20. Ketones	range(0, 8)	Integer
	21. Lactone	range(0, 1)	Integer
	22. N-Compounds	range(0, 7)	Integer
	23. Oximes	range(0, 1)	Integer
	24. Methoxy	range(0, 56)	Integer
	25. Oxirane	range(0, 1)	Integer
	26. Phenol	range(0, 5)	Integer
	27. Pyran	range(0, 3)	Integer
	28. Pyrazine	range(0, 2)	Integer
	29. Pyrrole	range(0, 1)	Integer

Table 7: Molecule-level features used for model input, their value ranges, and encoding types.

Feature Category	Features	Value Range	Encoding Type
	31. S-Compounds	range(0, 6)	Integer
	32. Sulfides	range(0, 3)	Integer
	33. Thiazoles	range(0, 1)	Integer
	34. Nitro	range(0, 3)	Integer
	35. Sulfonamide	range(0, 2)	Integer
	36. Thioesters	range(0, 1)	Integer
	37. Thiols	range(0, 4)	Integer
	38. Halogen	range(0, 5)	Integer
	39. Tert-butyl	range(0, 27)	Integer
	40. Nitrile	range(0, 2)	Integer

Table 8: Molecule-level features used for model input, their value ranges, and encoding types.

In this pipeline, feature extraction from molecular SMILES strings is implemented using the RDKit library for cheminformatics, combined with PyTorch for tensor construction and downstream machine learning. The pipeline extracts features at three levels: node-level (atom-level) features, molecule-level global features, and functional group counts. Together, these features comprehensively describe the structure, chemistry, and functional groups present in a molecule, providing rich input to graph-based learning models.

Node-Level Feature Extraction: For each atom in a molecule, the function constructs a feature vector that captures both numeric and categorical properties of the atom. RDKit provides per-atom properties such as atomic number, valence, formal charge, hybridization, aromaticity, and ring membership. These are encoded as integers, booleans, or enumerated integers as appropriate. These per-atom feature vectors are stacked into a matrix, one row per atom, and converted into a PyTorch tensor for model input. There are 15 node-level features per atom.

Global Molecular Feature Extraction: For molecule-level features, the function computes a set of global descriptors using RDKit descriptors such as molecular weight, logP (lipophilicity), topological polar surface area (TPSA), number of rings, number of rotatable bonds, hydrogen bond donors and acceptors, heavy atom count, formal charge, molecular complexity, and the length of the longest carbon chain. These 11 descriptors form the molecular-level (global) features and are encoded as floats or integers depending on their nature.

Functional Group Feature Extraction: Functional groups play a critical role in determining the chemical and sensory properties of molecules, making their detection highly relevant for odor classification tasks. In this pipeline, functional group detection is performed by matching molecular substructures against a predefined dictionary of SMARTS patterns, with each functional group (such as alcohols, amines, carbonyls, etc.) represented by a specific query pattern. Using RDKit's substructure matching methods, the pipeline identifies and counts the occurrences of these functional groups in each molecule. The resulting counts are aggregated into a structured feature vector, capturing the presence and abundance of chemically meaningful motifs that are often closely linked to specific odor characteristics. In total, 40 functional group features are extracted, providing interpretable domain-informed descriptors that contribute to accurate odor prediction.

C Model Development and Experimental Setup

We began by reproducing the baseline GCN model proposed in the baseline paper [1]. However, this reproduced baseline did not achieve comparable performance on our SoS (add refe to the git data link) dataset. This was due to the different dataset among baseline and SoS. To address the significant class imbalance observed in the data, we experimented with various loss functions, finding that the **focal loss** outperformed weighted cross-entropy by better emphasizing rare classes. We also explored a range of graph neural network architectures available in PyTorch, including **GraphConv**, **GAT**, and more complex architectures such as **GraphUNet**.

The GraphUNet architecture was designed to improve performance through a combination of graph pooling (SAGPool), skip connections, and an autoencoding objective that enhances subgraph explainability by identifying the most informative molecular substructures. Despite extensive tuning of this model, its performance did not surpass the baseline. Among all architectures tested, the **GIN** (Graph Isomorphism Network) layer consistently achieved the best results, with noticeable improvements in both recall and F1-score compared to the baseline GCN.

We also compared our results with the open-source implementation of OpenPOM, which was trained on the same dataset but reported significantly higher metrics. We attribute this difference partly to the use of **DeepChem**'s framework in OpenPOM, which provides higher-level abstractions and optimized training pipelines, as well as the use of RandomStratifiedSplit- a data splitting method that maintains balanced label distributions across folds. Since PyTorch does not natively offer this functionality, we implemented a similar stratification logic in our PyTorch pipeline, which led to further improvements.

#	Model Variant	AUROC	Precision	Recall	F1 Score
1	2 GATConv	0.845	0.332	0.475	0.358
2	4 SAGEConv	0.846	0.342	0.469	0.362
3	MPNN	0.860	0.327	0.319	0.324
4	Set2Set Readout	0.840	0.342	0.469	0.362
5	Use MuchPooling	0.860	0.303	0.554	0.323
6	Use DiffPool	0.835	0.576	0.267	0.281
7	GCN + SAGPool	0.845	0.394	0.335	0.342
8	GINE + SAGPool	0.842	0.306	0.416	0.348
9	One-Hot Encoding	0.8581	0.349	0.477	0.369

Table 9: Performance of different model variants and architectural modifications explored during model development.

Finally, we systematically documented the effect of each architectural choice, loss function, and training strategy on model performance, providing insight into the sensitivity of the model to different hyperparameters and highlighting the challenges of modeling on imbalanced, multilabel molecular odor datasets.