

# Olfactory Label Prediction on Aroma-Chemical Pairs

Laura Sisson

[laura.h.sisson@gmail.com](mailto:laura.h.sisson@gmail.com)

Aryan Amit Barsainyan

[aryan.barsainyan@gmail.com](mailto:aryan.barsainyan@gmail.com)

Mrityunjay Sharma<sup>1,2</sup>

[mrityunjay.csio24j@acsir.res.in](mailto:mrityunjay.csio24j@acsir.res.in)

Ritesh Kumar<sup>1,2</sup>

[riteshkr@csio.res.in](mailto:riteshkr@csio.res.in)

May 2024

<sup>1</sup> CSIR- Central Scientific Instruments Organisation, Sector 30-C, Chandigarh -160030, India

<sup>2</sup> Academy of Scientific and Innovative Research (AcSIR), Ghaziabad- 201002, India

## 1 Abstract

The application of deep learning techniques on aroma-chemicals has resulted in models more accurate than human experts at predicting olfactory qualities. However, public research in this domain has been limited to predicting the qualities of single molecules, whereas in industry applications, perfumers and food scientists are often concerned with blends of many molecules. In this paper, we apply both existing and novel approaches to a dataset we gathered consisting of labeled pairs of molecules. We present graph neural network models capable of accurately predicting the odor qualities arising from blends of aroma-chemicals, with an analysis of how variations in architecture can lead to significant differences in predictive power.

## 2 Introduction

Carefully designed fragrances and flavors appear everywhere in our daily lives, like in our food, drinks and hygienic products. However, designing fragrant molecules is a laborious and time consuming iterative process. The forefront of quantitative olfactory research has been the hunt for new and explicable features

used in the prediction of olfactory perception descriptors. The chemical space is huge (containing  $\approx 10^{60}$  molecules) [1], and ever-expanding.

Prior to the application of graph neural networks (GNNs) to odor prediction, researchers featurized aroma-chemicals based on specific molecular structures, like aromaticity and the presence of certain functional groups [2, 3]. These approaches achieved decent success on benchmarks like the DREAM Olfactory Challenge [3]; however, the adoption of GNNs to this domain led to significant improvements in the predictive power of contemporary models [4]. Instead of hand-engineered featurizations, recent methods for searching and analyzing the chemical space have utilized data-intensive stochastic or deep learning methods [5, 6]. New machine learning architectures which operate on graphical or plain text representations of molecules have recently emerged [7, 8, 9, 10], leading to improvements in computational understanding across drug discovery, material development, molecular property prediction, and de novo molecular design [11, 12, 13, 14]. Recently, Lee et al. [15] used GNNs to predict aroma labels with high accuracy and precision, building a "perceptual odor map" from the underlying vector-embedding representations for each molecule.

These breakthroughs in odor label prediction have enabled researchers to gain deeper insight into the relationship between scent and molecular structure. However, the non-linear relationships occurring in mixtures of aroma-chemicals have yet to be untangled. We adapt these deep learning methods and also apply new techniques to generate embeddings for blends of aroma-chemicals. Given the predominantly proprietary nature of research utilizing GNNs in chemistry, we needed to explore a variety of architectures to obtain these results.

This work represents a natural progression from the single molecule prediction task, where datasets of labeled molecules are individually processed, to a new blend-focused domain, where models are capable of predicting the odor of mixtures of molecules.

The contribution of this work is two fold: firstly, we present a carefully compiled odor mixture dataset wherein the perceptual descriptors of individual components and their blends are available; secondly, we present a set of publicly available GNN based algorithmic frameworks to predict the labels of the mixtures. We also present a number of selected experiments which showcase the efficacy of our approach, where our GNNs can transfer underlying models from blend prediction to single molecule prediction and vice versa [16].

## 3 Methods

### 3.1 Data Collection

To build the blended pair dataset, molecular structures (SMILES) and odorant labels were gathered from the GoodScents online chemical repository [17] to generate a dataset of aroma-chemical blends. While the GoodScents website cataloged ~3.5k molecules, each aroma-chemical's page suggested a complementary odorant (called a "blender") that, when mixed together, yielded distinct

aromas. Each molecule’s page often contained more than 50 such recommendations, enabling us to gather over 160k molecule-pair data points

We built an adaptive web crawler that parsed the names, olfactory labels and recommended blenders for all odorants across GoodScents using the Python package BeautifulSoup. Non aroma-chemicals odorants were filtered out. In instances where aroma-chemicals lacked SMILES entries, or where the SMILES was unable to be parsed such data was excluded. These malformed data points made up only 0.05% of the total labeled pair dataset, and were therefore removed.

The collection of all molecule pairs in the database forms a meta-graph, where each node is itself a molecular graph, with edges between nodes if there are odor-labels for the blended pair of molecules. In order to ensure train/test separation, the meta-graph was carved into two components with the following requirements: firstly, in order to prevent distribution shift, each component must contain blended pair data points covering every label; secondly, in order to maximize the amount of usable data, the number of edges between the components (known as the edge-boundary degree) should be minimized, as these data points are thrown out to ensure train/test separation. Efficiently minimizing the edge-boundary degree is NP-hard in the general case [18], although previous research demonstrates that some special cases can be solved in polynomial time [19].

### 3.2 Dataset

The dataset generated contains discrete labels for 109 olfactory notes<sup>1</sup>, which we standardized. Some entries contained pairs with no labels (marked as ‘No odor group found for these’) and these non-labeled pairs were removed. Additionally, “anisic” was substituted with the more common “anise”, “medicinal,” (with a trailing comma) was adjusted to “medicinal”, and “corn chip” was replaced with “corn”. These modifications led to a final set of 104 notes.

To examine the transfer learning capabilities of our models, we derived the single odorant datasets from Leffingwell [20] and GoodScents [17], which were available as a combined dataset [21].

---

<sup>1</sup>There was no data available for relative concentrations in the blends.

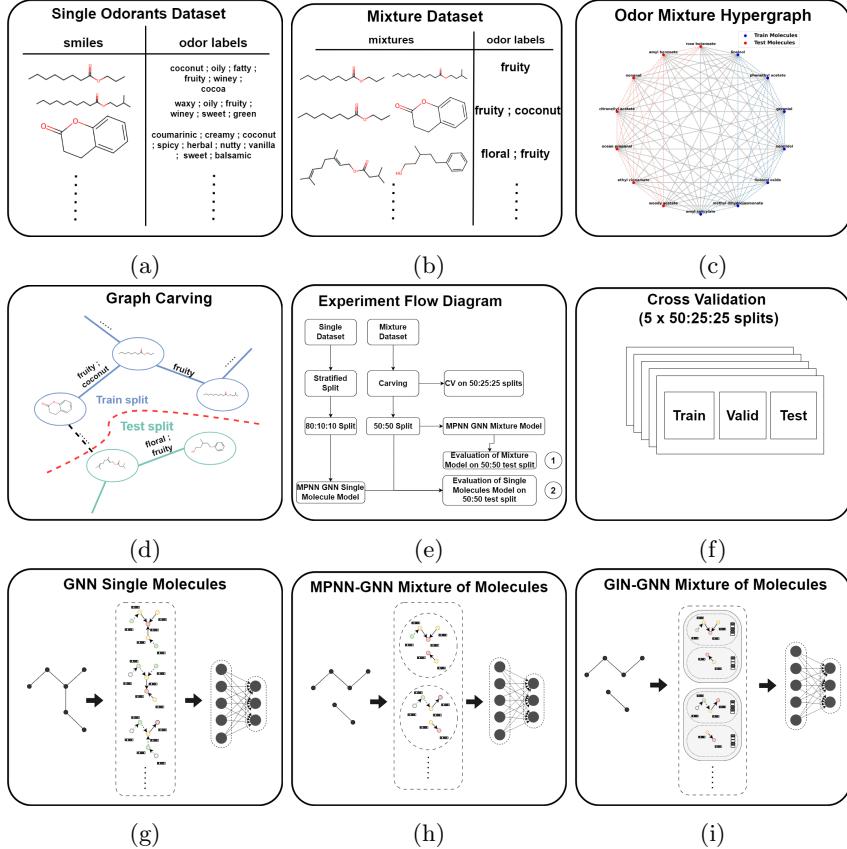


Figure 1: **Methodology**

**(a and b)** Non-linear relationship between the qualities of constituent aroma-chemicals and the overall blend. Although the same molecules appear across the single and mixture datasets, emergent notes appear when molecules are combined, and other notes become muted in the blend.

**(c)** Sample of the densest region of the blended pair meta-graph. Here, 0.5% of the total meta-graph nodes, consisting of 7 train molecules (in blue) and 7 test molecules (in red) are visualized, with an average degree of 6. Because there are many data points/edges per molecule, the meta-graph is dense and thus difficult to carve.

**(d) Graph carving schematic.** The carving algorithm aimed to maximize the number of usable pairs without causing distributional shift in labels.

**(e & f) Schema of experiment.** The entire (e) optimization and training pipeline used in this paper, including (f) 5-fold splits of 50:25:25 train/test/validate splits used for hyper-parameter optimization.

**(g) GNN predictions on single molecules.** Message passing layers are applied across the molecular graph, and then followed by a readout phase and a multi-layer perceptron (MLP) in order to predict the final label.

**(h) MPNN-GNN predictions on blended pairs,** the molecular graphs are treated as one graph, with a combined readout and MLP as above.

**(i) GIN-GNN predictions on blended pairs.** The molecule graphs have separate message passing and readout steps, and are combined only before the MLP.

### 3.3 Graph Carving

We carved the meta-graph by randomly partitioning the set of molecules into train and test splits. The carving algorithm was repeated until we generated a carving with at least one train and test data point for every label. With a 50:50 split, this took 90k iterations. We experimented scoring potential graph carvings using the Kullback–Leibler divergence between the train/test odor-label distribution and the distribution across the entire graph. However, we decided to optimize the number of usable data points rather than the similarity between components. Raising the fraction of training data meant that less data points were discarded, but also meant that it became impossible to carve a test set of molecules which covered all labels. Therefore, the 50/50 split was selected. This resulted in a final dataset with 44k training pairs and 40k test pairs, discarding 83k data points to satisfy the separation requirements. Out of the 109 odor labels, only 74 appeared across enough molecules to be carved. We attempted 250k further carvings, but never one that covered more labels.

### 3.4 Cross Validation

In order to test and identify the best performing models, we performed various sets of experiments on the carved train and test components of the datasets. These experimental procedures are documented in Fig. 1.

1. **5-fold split and hyperparameter search:** We conducted further searches to obtain five train-test-split carvings, which were fully separated as above. Because each component had less constituent molecules, the number of odor labels that were possible to carve was only the range of 41 to 44 per fold. We used these 5-fold splits to conduct hyperparameter optimization, hypothesizing that the best models would transfer well to the original 74 label 50:50 carving.
2. **50:50 split on the model based on mixture data:** In this experiment we trained 3 MPNN-GNN models, with different random seeds, on the original 50:50 train-test split using the best hyperparameters found.

## 4 Architecture

### 4.1 Models

We trained a variety of GNN models for predicting the blended odor labels from the structures of pairs of aroma-chemicals. These models were derived from two primary architectures.

Firstly, we developed a novel Graph Isomorphism Network (GIN) [22] based model, which independently generated embeddings for each molecule in a pair. These embeddings were combined during a final blended-pair prediction step. For detailed analysis of the GIN-GNN model, see section 1.2.1 of the supplementary information.

Secondly, we trained several variations on the Message Passing Neural Network (MPNN) inspired by Lee et al [15]. Here, the molecular structures for pairs of molecules were grouped into a single graph before being fed into the message passing layers. For the detailed analysis of the MPNN-GNN model, refer section 1.2.2 of the supplementary information.

## 5 Results

### 5.1 Blended Pair Prediction

To measure the predicting powers of the various models, we used the area under the receiver operating characteristic curve (AUROC) for each odor label. To compare the results, we took the micro-average across all test data points. The MPNN-GNN model achieved a mean AUROC of 0.77, with the GIN-GNN scoring a slightly lower mean AUROC of 0.76. For context, a naive 0-R model using the mean frequency of each label across all molecules as a constant prediction achieves an AUROC of 0.5 for every label, by definition. As a baseline model, we generated 2048-bit Morgan fingerprints (MFP) with  $radius = 4$  for each molecule in a pair, which were concatenated and then used as input to a logistic regression to predict the blended pair’s labels.

While the GIN-GNN model predicts some labels very accurately, it significantly underperforms compared to the random baseline on other labels. On the other hand, the MPNN-GNN model performs consistently across all labels. One of the easiest descriptors to predict was “alliaceous” (garlic), reflecting previous work [15] which suggested that this note straightforwardly correlated with the presence of sulfur in the molecule. Unlike in previous work, our models accurately predicted the label “musk”, which is normally difficult to predict as it occurs across many different structural classes of molecules. However, direct comparison between benchmarks is not straightforward, as previous work predicted continuous ratings for odor, and our dataset contains discrete labels. There were some descriptors (like “orris” and “earthy”) which none of the molecules were able to accurately predict.

### 5.2 Single Molecule Prediction

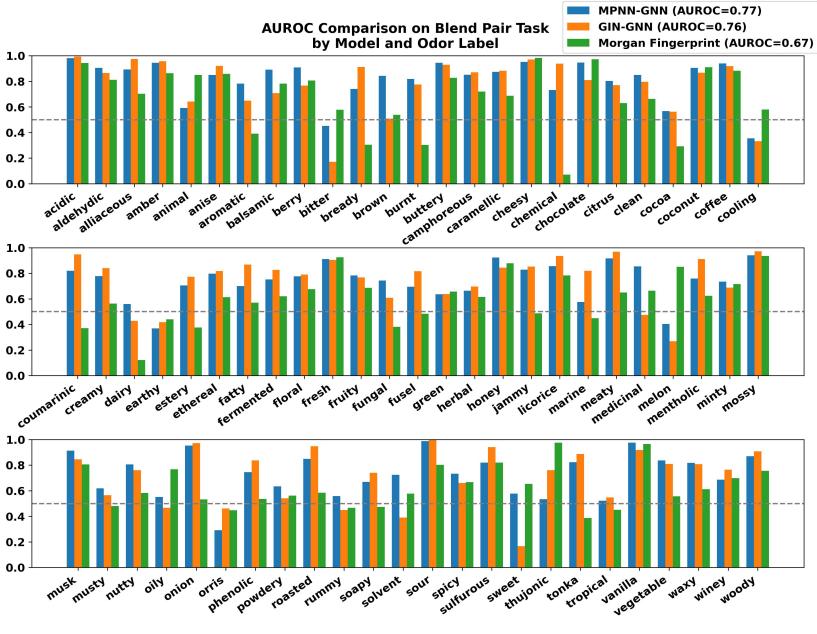
We also measured the performance of our models on the single molecule prediction task. To adapt the GIN-GNN model to this task, we generated graph-level embeddings for each molecule and trained a logistic regression classifier to predict the same 74 odor labels. Because the graph-level and original pair-level embeddings were of different dimensionalities, the MLP portion of the architecture could not be transferred. For the MPNN-GNN, the only modification necessary was inputting a single molecule instead of a pair of molecules into the message passing phase. The entire trained architecture was able to be reused. On the single molecule task, the MPNN-GNN achieved a mean AUROC score of 0.89, while the GIN-GNN and Morgan Fingerprint models achieved scores of

0.85 and 0.82 respectively.

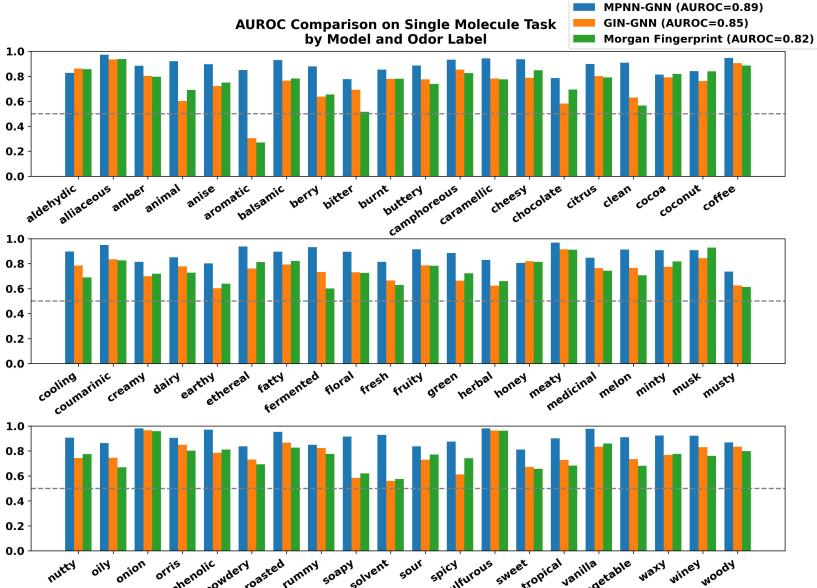
The significant improvement of all models on the single molecule prediction task, as compared to the blended pair task suggests that the former task is much harder than the latter. We hypothesize the widened gap between the MPNN-GNN and the GIN-GNN on this task is due to the fact that the GIN-GNN’s prediction layers could not be reused.

Regardless, the descriptor “alliaceous” remained easy to predict across the board, but surprisingly, “musk” was one of the easiest descriptors to predict. In our dataset, “musk” molecules, regardless of structural class, were often combined with each other in blends. This likely produced similar embeddings in our GNN models for the molecules, even though they were structurally dissimilar. This provided an advantage over previous work, where models had to learn the olfactory similarity for “musk” molecules from their labels alone. One notable exception was the label “aromatic”, which the GIN-GNN and Morgan Fingerprint models failed to accurately predict.

The best hyperparameter values derived from experiments conducted on the combined Leffingwell-GoodScents dataset for both the blended pair and the single molecule trained model tasks are detailed in Tables S2 and Table S3 in the supplementary information.



(a)



(b)

Figure 2: Predictive power of our GNN models and the Morgan Fingerprint baseline across all labels with random baseline (dashed line).

(a) Blended pair task AUROC scores, by descriptor.

(b) Single molecule task AUROC scores, by descriptor.

## 6 Discussion

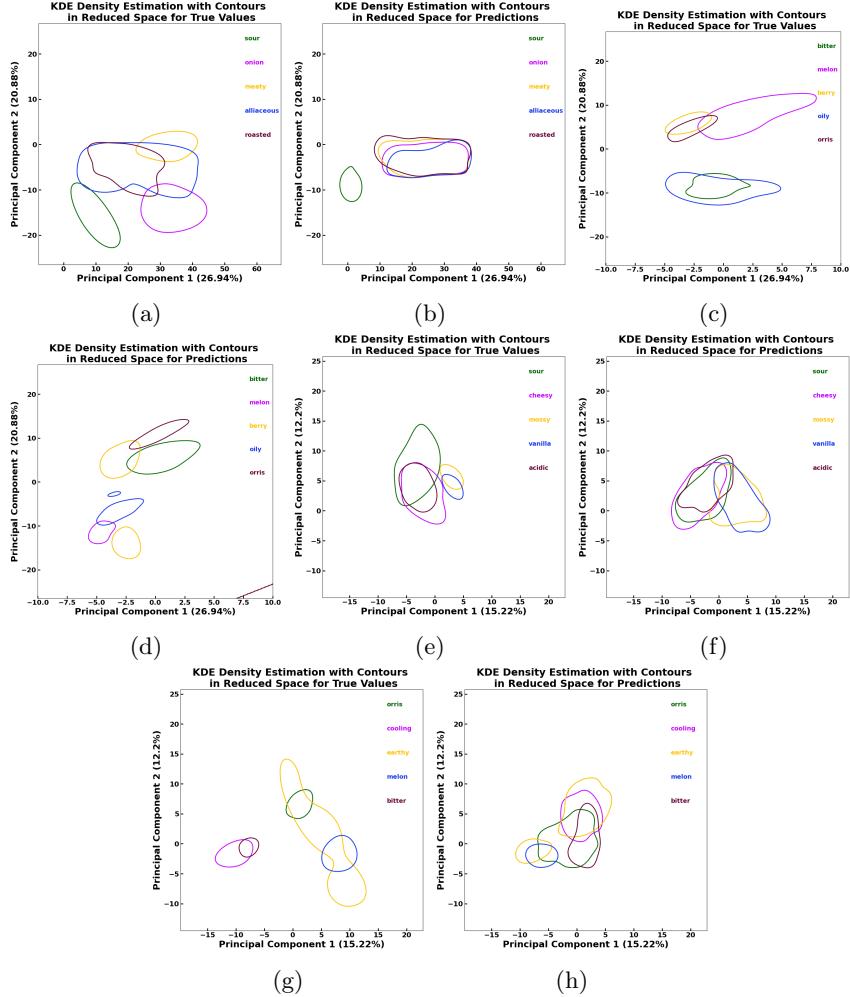
### 6.1 MPNN-GNN Performance

Several experiments were conducted to understand the MPNN-GNN’s performance on individual labels, focusing on the best and worst performing odor labels. Our goal was to analyze how model predictions correlated with the co-occurrence patterns of odor descriptors in blended pairs across both the training and test sets. Figure 3 shows the kernel density estimation (KDE) plots of the MPNN-GNN’s most accurate and least accurate predictions on both the single molecule and blended pair prediction task, alongside the ground-truth values.

Interestingly, the MPPN-GNN’s predictive scores for the top 5 and bottom 5 are not correlated with the descriptors’ occurrences in the training data, contradicting the findings of Lee et al. [15]. We conducted detailed analysis for a small set of descriptors:

- “**orris**”: Though “orris” has significant representation in the training and test sets, an imbalance in the data favors “spicy” (both individually and in co-occurrence with “orris”), potentially causing the model to prefer “spicy” as the primary descriptor for mixtures containing “orris”.
- “**cooling**”: In the test set, 75% of the “cooling” target distribution is also labeled “green”, but the model predicts 50% as “estery” and 30% as “fruity”.
- “**green**”: The MPPN-GNN predicts “green” as occurring in less than 2% of test molecules, despite the label’s major presence in the training and test sets. This could again be attributed to the individual co-occurrence of “green” label with “fruity” and “estery” in the overall database.
- “**earthy**”: Similarly for “earthy” labels, the model predicts 40% as “fatty” and 30% as “chocolate”, despite many “earthy” pairs occurring in the training and test sets. “Waxy” is well-predicted for mixtures smelling both “earthy” and “waxy”.

There seems to be semantic spillover across labels with regards to their co-occurrences in both the blended pair and single molecule tasks. To a certain degree, this is expected, as olfactory labels are often nebulous and overlapping. To remedy this, further research into a definitive olfactory vocabulary must be conducted, where all descriptors are semantically distinct from each other [15, 23]. For further detailed analysis of the performance across labels, see “MPNN-GNN Performance” in the supplementary information.



**Figure 3: Analysis of Odour labels by conducting experiments.**

(a) **KDE plots for top 5 descriptors** by predictive accuracy in the training set of single molecule task.

(b) **KDE top 5 single molecule as above**, for test set.

(c) **KDE plots for bottom 5 descriptors** by predictive accuracy in the training set of single molecule task.

(d) **KDE bottom 5 single molecule as above**, for test set.

(e) **KDE plots for top 5 descriptors** by predictive accuracy in the training set of blended pair prediction task.

(f) **KDE top 5 blended pair as above**, for test set.

(g) **KDE plots for bottom 5 descriptors** by predictive accuracy in the training set of blended pair task.

(h) **KDE bottom 5 blended pair as above**, for test set.

## 6.2 GIN-GNN Performance

Across both the blended pair and single molecule prediction task, the GIN-GNN underperforms compared to the MPNN-GNN. We hypothesize this is primarily due to a number of differences in the architectures.

Firstly, molecules are fed into the GNNs in different ways. As stated above, the MPNN-GNN applies the message passing function across the pairs of molecules as if they were a single graph, with two disconnected components. The readout layer combines all the nodes across both graphs simultaneously. From there, additional feedforward layers generate the prediction, and this feedforward neural network can be reused for the single molecule prediction task. On the other hand, the GIN-GNN generates component embeddings separately and concatenates them together prior to the final feedforward layers. We hypothesize that this leads to weaker underlying representations for component molecules in the GIN-GNN, as shown in the performance comparison in Figure 2a and 2b, the difference is especially noticeable for the single molecule prediction task, where a new feedforward network must be trained from scratch.

We also propose that the additional information provided by edge-conditioning in the MPNN-GNN results in a higher predictive capacity than in the GIN-GNN, which updates the hidden states of nodes based only on the node’s own features.

Finally, we note that the set2set readout method provided better results than simply concatenating the mean and max pooling results of the node level embeddings.

## 6.3 Embedding Space

Although the relationship between the odors of individual molecules and the blended pairs in which they occur is non-linear, previous work has shown that embedding spaces can represent non-linear relationships with linear transformations.

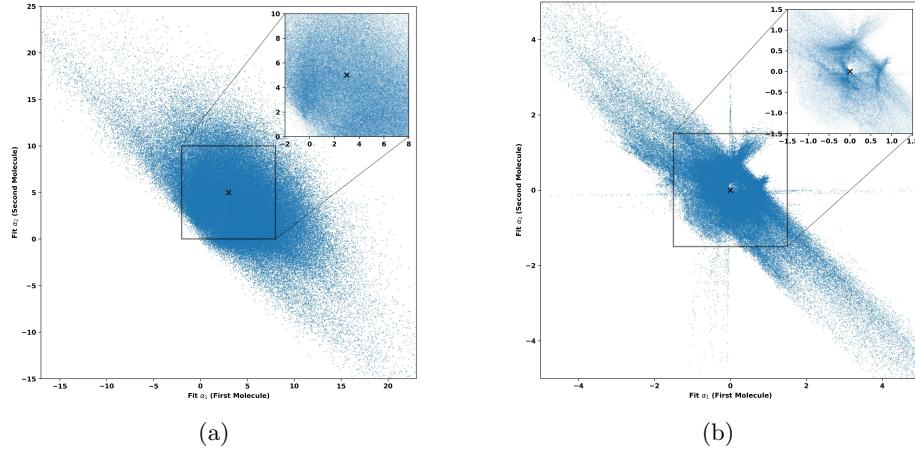
In order to explore the relationships between the latent space of individual molecules and that of the blended pairs, we generated three vector embeddings per data point:  $\mathbf{e}_1$  (the embedding of the first molecule in the pair),  $\mathbf{e}_2$  (the embedding of the second molecule in the pair), and  $\mathbf{e}_p$  (the embedding of blended pair), using the MPNN-GNN. To determine if the blended pair embeddings were simply linear combinations of the constituent embeddings, we fit a number of linear regression models, one per pair in the dataset:

$$\alpha_1 \cdot \mathbf{e}_1 + \alpha_2 \cdot \mathbf{e}_2 = \mathbf{e}_p$$

where  $\alpha_1$  and  $\alpha_2$  are constant coefficients. We measured how well the linear regression model fit the relationship using the coefficient of determination ( $r^2$ ). Across all blended pair data points, the average ( $r^2$ ) was 0.47. While there was a significant portion of the relationship between constituent and blended pair embeddings that could be explained linearly, the majority was more complex. Notably,  $\alpha_1$  and  $\alpha_2$  were inversely correlated, meaning that while a significant portion of blended pair embeddings were equal combinations of each

constituent molecule, as the influence of one molecule increased in the blend, the other molecule’s influence decreased. This may be explained by the phenomena of additive or subtractive blending, wherein some molecules intensify the certain notes when combined, while other molecules may mute each other in combination.

The process was repeated with the GIN-GNN, though the component embedding space ( $D=100$ ) and the pair embedding space ( $D=77$ ) could not be directly compared, so we reduced the component embedding space to the same dimension as the pair embedding space, and fit linear regressors as above.



**Figure 4: Scatter-plots of fit-coefficients for predicting the blended pair’s embedding using the GNN embeddings.**

**(a) Scatter-plot of the fit coefficient using the MPNN-GNN model from,** with zoom on centroid. Across all pairs, the average  $r^2$  is 0.47 and the  $p$ -value for the  $F$ -statistic is  $4.68 \times 10^{-5}$ . Notably, the distribution is not centered on the origin. In some cases, the blended pair’s embedding consists of equal combinations of each individual embedding, while in other cases, one particular embedding predominates.

**(b) Scatter-plot, as above, using the GIN-GNN embeddings.** The correlation between single molecule and blended pair embeddings was weaker, with average  $r^2$  of 0.021 and with a  $p$ -value for the  $F$ -statistic is .445. The distribution is centered on the origin, suggesting that for many points, neither molecule’s embedding factor into the pair embedding. The vertical and horizontal lines represent where one component predominates, but the other molecule is not factored in at all.

## 7 Conclusion

By applying deep-learning techniques to a novel dataset, we trained a set of models capable of accurately predicting the non-linear olfactory qualities of

aroma-chemical blends. Our GNNs transfer strongly to single molecule prediction tasks. They are available on GitHub for further exploration.

In our opinion, the ultimate research goal in this domain is to produce a model capable of predicting continuous labels for blends of many aroma-chemicals at varying concentrations. This mirrors the real-life work done by food scientists and perfumers.

Well-labeled public olfactory datasets that would enable this research remain scarce even for the single molecule case. Though fragrance companies likely have extensive libraries of blend recipes, these datasets remain proprietary. This paucity of public data necessitates the application of innovative techniques to boost model performance and also the exploration of novel strategies for dataset augmentation. Our work stands as a proof of concept for further research in these domains.

### 7.1 Acknowledgements

We thank Dr. Andreas Keller for mentorship and technical guidance.

# Supplementary Information: Olfactory Label Prediction on Aroma-Chemical Pairs

## 1 Methods

### 1.1 Note Canonicalization

For labels that are difficult to predict because of their appearance across multiple structural classes, researchers and perfumers may benefit from using different labels specific to each structural class. “Musk” is one such label and though “floral musk” and “soft musk” are frequently used to distinguish between different kinds of “musky” odors, difficulty arises when “musk” is used directly as a note, instead of as a family of notes. Future work could determine the feasibility of splitting this note/family apart into a number of distinct odor words. Researchers could task a panel of experts to determine if two molecules, both labelled “musk”, come from the same or different structural class. If “musks” from different classes are easily separable, then new descriptive words are called for.

In the same vein, our paper uses 74 labels out of a set of 109 descriptors, simply based on availability. There is no agreed on canonical set of odor descriptors, and previous works have used sets 138 labels[15], 131 labels[24], and as few as 19 labels[3]. Though it is possible to predict ratings on one set of labels from another[25, 26], a canonical set would allow direct comparison between different approaches.

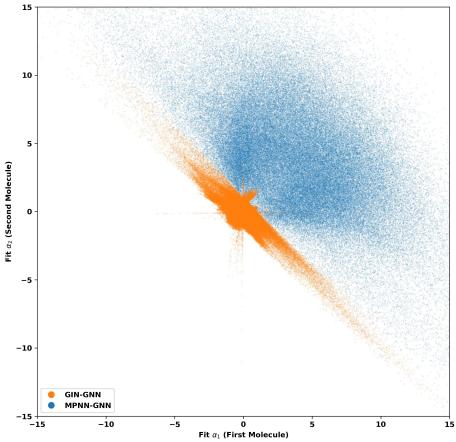
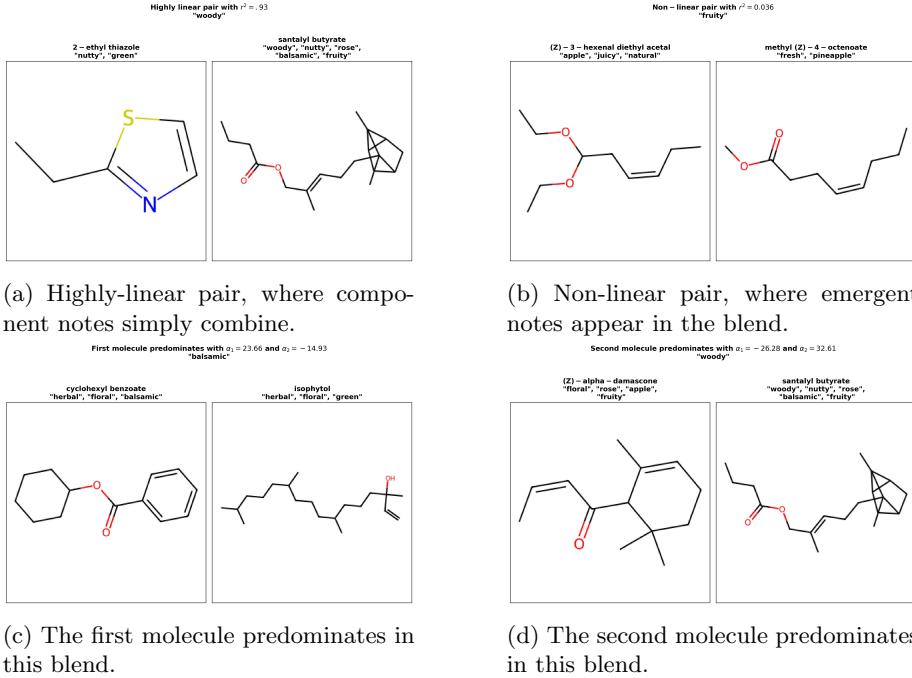


Figure S1: Combined scatter-plot of the regressors for both the MPNN-GNN and the GIN-GNN.



**Figure S2: Four pairs of molecules, representing different blending outcomes.** While the blended results of some pairs are highly-linear, and thus the notes in the blend are easily predicted from the component molecules, other pairs are non-linear and difficult to predict. The balance between each component is also notable, as in some blends the first molecule predominates, while in others the second molecule is much more apparent.

As shown in the Table S6 there were 11 notes that were strict subsets of other notes i.e. they exclusively linked to a parent note. While there were 3 notes as shown in Table S7 that existed independently .

The relationship between the odors of blends and their constituent molecules is intricate, as evidenced numerically by the low Jaccard coefficient observed between combinations of constituent labels and blend labels. Utilizing the set union of the labels of the constituent molecules yielded a Jaccard coefficient of 0.18, while employing the set intersection resulted in a Jaccard coefficient of 0.27. Clearly, linear prediction of blended labels from constituent labels is not feasible.

## 1.2 Models and Hyperparameter Search

Hyperparameter optimization is a challenging task, but research papers rarely provide the final hyperparameters used to train their models, and even fewer

fully enumerate their search space. Much of the hyperparameter optimization procedure used here was adapted from “Neural Message Passing for Quantum Chemistry”[27] in which researchers built a GNN model for predicting quantum properties of organic molecules. These hyperparameters transferred well to our dissimilar task/GNN architecture, and are likely useful across disparate deep-learning chemistry domains.

### 1.2.1 Graph Isomorphism Model

We tested a variety of architecture using Optuna[28] for our hyperparameter search to find an initial, bespoke blended pair prediction model. The best architecture was structured as follows:

The Graph Isomorphism Network (GIN)[22], built in PyTorch Geometric[29], was selected as the GNN architecture. The GIN was used for three message passing steps, and in order to allow weight-tying between these steps, the initial Open Graph Benchmark[30] atomic encodings were padded to the hidden layer dimension ( $D = 832$ ). The carving algorithm was repeated until we generated a carving with at least one train and test data point for every label. With a 50/50 split, this took 90k iterations. Each graph carving was experimented with using Kullback–Leibler divergence, which considers the disparity between the train/test odor-label distribution and the distribution across the entire graph. However, the decision was made to prioritize optimizing for the number of usable data points rather than the similarity between components. Raising the fraction of training data meant that less data points were discarded, but also mean that it became impossible to carve a test set of molecules which covered all labels. Therefore, the 50/50 split was selected. We used a two layer feedforward neural network, built in PyTorch[31], as the update function.

The GIN generated embeddings for every atom/node in both molecules across the pair. The node-embeddings were combined using global mean and add pools concatenated together to generate graph-level embeddings, for each molecule in the pair. The graph-level embeddings for the two molecules were then further concatenated (in arbitrary order) and passed through another two-layer feedforward network to generate the pair-level embedding (also of  $D=832$ ). This arbitrary ordering of graph-level embeddings worked well for the molecule-pair task, as molecules tend to appear evenly between the first and second position in the pair as a by-product of the graph-carving, but for blends of 3+ molecules, more advanced techniques, like a Set2Set[32] model would be needed to combine the graph-level embeddings. Logits for all 74 odor labels were predicted linearly from the pair-level embeddings. We used a binary cross-entropy loss for the predicted labels with respect to their true values. The model was scheduled for 250 epochs, but it was terminated after 121 epochs using early-stopping ( $patience = 0$ ). We used the Adam optimizer ( $lr = 2.1e-5$ ) with a decaying learning rate ( $decay = 0.08$ ) across the first 90% of training epochs. The entire hyperparameter search space was as follows:

Table S1: Hyperparameter values chosen as per procedure adapted in “Neural Message Passing for Quantum Chemistry” [27]

Hyperparameter	Search Space	Chosen Value
Training Epochs	[100,500]	250
Initial Learning Rate	[1e-4,1e-1]	2.1e-5
Learn Rate Decay	[1e-5,1e-1]	2.1e-5
Hidden Dimension	[32,1024]	832
Feedforward Layers	[1,6]	2
Message Passing Steps	[1,13]	3
GNN Architecture	{ <i>GIN,GCN,MPNN</i> }	<i>GIN</i>
Aggregation	{ <i>Set2Set,Mean+Add Pool</i> }	<i>Mean+Add Pool</i>

### 1.2.2 MPNN-GNN

We used the MPNN-GNN model[15], which was originally trained on 138 odor descriptors using LF-GS combined dataset [2] to predict odors of single molecules. The model contains message passing layers, a readout layer with radius 0 combination to fold atom and bond embeddings together followed by set2set operation and a feed-forward network layer with sigmoid activation for multi-label classification predictions.

A summed binary cross-entropy loss is used in the training process for every descriptor. Each descriptor’s influence is taken into account in the loss computation, and its weight is determined by a factor of  $\log(1+\text{Imbalance ratio per label})$ . For every label  $\lambda$  in a multi-label dataset M, the Imbalance ratio per label (IRLbl) is calculated, where  $Y_i$  is the label-set of the  $i^{th}$  instance. The ratio of the specific label  $\lambda$  to the majority label is used to calculate IRLbl. For the most common label, it takes on a value of 1, while for other labels, it takes on a greater value. The imbalance for the associated label increases with the IRLbl value.

The hyperparameter search space is given in Table S2.

Table S2: Best hyperparameter values chosen from experiments on the LF-GS dataset for mixture model odor prediction

Hyperparameter	Search Space	Chosen Value
ffn Dropout probability	[0.12,0.25,0.5]	0.12
Weight decay	[1e-3,1e-4, 1e-5 ]	1e-5
Initial learning rate	[0.001,0.0001, 0.005,0.0005]	0.001
Decay Rate	[0.25,0.5,0.75]	0.5
Decay Steps	[42 * 5,42 * 10, 42 * 15, 42 * 20]	42 * 20
Number of step set2set	[2,3, 4]	3
Number of layers set2set	[2,3,4],	3
FFN hidden list	[[200], [60, 60], [300], [500, 500], [300, 300]]	[300]
Aggregation	{ <i>global sum pooling,set2set</i> }	<i>set2set</i>

### 1.2.3 Single Molecule Trained Model

In this study, we aim to forecast the odors of molecular mixtures. Our dataset of odorant mixtures includes 74 odor notes, with 60 of these notes also present in the LF-GS combined dataset. In the experiment with the model trained on single molecules, we utilize the LF-GS dataset for training and test it on a 50:50 split of the mixture dataset. Given that the odorant mixture dataset was derived from Goodscents data, part of the LF-GS combined dataset, we remove common SMILES from the test set in the LF-GS dataset. Consequently, 60 out of 138 odor notes are used. This subset of the LF-GS dataset is then divided into training and validation sets in an 80:20 ratio using random stratified splitting. We proceeded with using the best parameters from experiments on the LF-GS

Table S3: Best hyperparameter values chosen from experiments on the LF-GS dataset for single molecule trained model prediction

Hyperparameter	Search Space	Chosen Value
ffn Dropout probability	[0.12,0.25,0.5]	0.25
Weight decay	[1e-3,1e-4, 1e-5 ]	1e-5
Initial learning rate	[0.001,0.0001, 0.005,0.0005]	0.005
Decay Rate	[0.25,0.5,0.75]	0.75
Decay Steps	[4 * 5,4 * 10, 4 * 15, 4 * 20]	4 * 10
Number of step set2set	[2,3, 4]	2
Number of layers set2set	[2,3,4],	2
FFN hidden list	[[200], [60, 60], [300], [500, 500], [392, 392]]	[200]
Aggregation	{ <i>global sum pooling</i> , <i>set2set</i> }	<i>set2set</i>

dataset for single molecule odor prediction. (Note: The model hyperparameters could have been fine tuned on 100 trials of random search)3 models with different random seeds were trained to predict odors of single molecules. Then the models were tested on Test split from odor mixture dataset to assess how the model performs on predicting odors of mixture of molecules.

We observed that the mean ROC score for Test split (average of 3 models) is  $0.8029 \pm 0.00147$  (95% confidence level). The ensemble of these 3 models gave ROC score for Test set = 0.8146. We have 5 folds of train-valid-test split of the odor mixture set with odor notes in the range of 41 to 44 per fold. The assumption is that the best hyperparameters from that run would show similar results with 50:50 split with 74 odors. The reason we had to do this was the difficulty in extracting 5 fold train-valid-test splits with 74 odors notes, due to the limitation of the graph carving and nature of the mixture datasets. We performed 100 trials of random search cross validation to get the best values for hyperparameters like regularization weight decay, set2set readout parameters, dropout and hidden layer size of feed forward layer, initial learning rate, decay rate and decay step for exponential decay learning rate scheduling. The best validation ROC score is 0.7928 and best test ROC score is 0.7806.

**Note:** We used the best hyperparameters for message passing layers for

MPNN-GNN from single molecule training results.

## 2 Supplementary Data

Table S4: Top 10 predicted labels by the model

Label	MPNN-GNN	MPNN-GIN	Morgan Fingerprint
acidic	0.98	0.99	0.94
cheesy	0.95	0.97	0.98
mossy	0.94	0.97	0.94
sour	0.99	1.00	0.80
vanilla	0.98	0.92	0.97
amber	0.95	0.96	0.86
coffee	0.94	0.92	0.88
buttery	0.95	0.93	0.83
fresh	0.91	0.90	0.92
chocolate	0.95	0.81	0.97

Table S5: Bottom 10 predicted labels by the model

Label	MPNN-GNN	MPNN-GIN	Morgan Fingerprint
dairy	0.56	0.43	0.12
earthy	0.37	0.42	0.44
orris	0.29	0.46	0.45
cocoa	0.57	0.56	0.29
rummy	0.56	0.45	0.47
tropical	0.52	0.55	0.45
bitter	0.45	0.17	0.58
cooling	0.35	0.33	0.58
musty	0.62	0.57	0.48
sweet	0.58	0.17	0.65

## 3 Results

### 3.1 MPNN-GNN Performance

To understand the performance of the MPNN model at the level of individual labels, we carried out a series of experiments. In this study, we investigate the targets of the worst and best performing odor labels, as well as the actual predictions made by the model. We aim to analyse how model predictions correlate with odor descriptors of individual components within the mixture, both in the training and test sets.

Table S6: Notes that occurred exclusively in association with a parent note

Dependent Note	Parent Note	Frequency
estery	fruity	739
cherry	floral	443
toasted	butterfly	218
juicy	sulfurous	185
tomato	vegetable	138
tobacco	nutty	101
potato	vegetable	51
celery	lactonic	34
lactonic	celery	34
dusty	fruity	32
tarragon	anise	28

Table S7: Isolate Note with the frequency

Isolate Note	Frequency
ammoniacal	9
salty	1
hay	2

The inference that we could draw from the experiments was that the top and bottom 5 performances are not correlated to training data size. While Lee et al. [15] observed a notable improvement in model performance with an increased number of training examples, our study did not replicate this finding. The insights derived from analyzing the results of the odor mix model for the bottom five predicted labels are as follows: **a) Orris:** Our analysis revealed that 20% of “orris” samples in the test set also have “spicy” as a label. Despite significant representation of “orris” in both the training and test sets, there’s a notable imbalance: 42 “orris” samples in the training set compared to 1238 “spicy” samples. This imbalance could potentially cause the model to favour “spicy” as the primary descriptor for mixtures containing “orris”. **b) Cooling:** In the test set, approximately 75% of the “cooling” target distribution is “green” and less than 20% is “fruity”. However, the model predicts about 50% as “estery” and 30% as “fruity”. Despite “green” being a major descriptor in both the training and test sets, predictions for “green” are less than 2% for mixtures with a “cooling” target. The model appears biased towards component odors not present in the training set mixtures. **c) Earthy:** For the “earthy” target distribution, around 20% are “musty” and “waxy” in the test set. However, predictions vary: 40% are predicted as “fatty”, 30% as chocolate, and only a small share as “waxy” or musty. “waxy” is well-predicted for mixtures that smell both “earthy” and “waxy”. Both the training and test sets have high “earthy” content, indicating no bias. **d) Melon:** For the “melon” target distribution,

there is a very small percentage of “floral” and “fruity” in the test set. Model predictions are poor, with over 40% predicted as “creamy”. Both the training and test sets have high “melon” content, indicating no bias. **e) Bitter:** For the “bitter” target distribution, there is a small percentage of “waxy” and “fatty” in the test set. The model most often predicts cheesy and sour. The test set has higher cheesy content, with no components smelling “bitter”, indicating that “bitter” might be an emergent odor. The model seems biased towards component odors not in the training set mixtures. Less than 50% of the training set mixtures have a “bitter” odor in any component.

From the analysis of the odor mix model, the top five predicted labels provide the following insights: “Sour”, “acidic”, “vanilla”, “mossy”, and “cheesy” perform as expected, with the highest prediction scores for their respective odor labels. Even their mixture components exhibit a high percentage of these odor labels.

The inferences drawn from the analysis of the odor single model results on the mixture test set for the five bottom predicted labels are: **a) Bitter:** For the “bitter” target distribution, the test set has a small percentage of “waxy” and “fatty”. However, the model predicts “sour”, “winey”, and “buttery” with the highest score. None of the mixture components in the test set have a “bitter” smell, explaining the poor performance of “bitter” with a ROC score of < 0.1. The model seems to predict mixture odors based on individual components. **b) Melon:** For the “melon” target distribution, the test set has minimal “floral” and “fruity” percentages. Model predictions for “melon” mixtures are quite poor, with little more than 50% predicted as coumarinic. The training (single molecules) and test mixture components have high “melon” content. **c) Berry:** For the “berry” target distribution, the test set has a small percentage of “floral” and “fruity”. Model predictions for “berry” mixtures are quite poor, with over 35% of predictions not assigning any odor label. Interestingly, the training (single molecules) and test mixture components have high “berry” content. **d) Oily:** For the “oily” target distribution, the test set has minimal “fruity”, “ethereal”, “green”, and rummy odors. Model predictions are poor, with over 25% not assigning any odor label. Interestingly, mixtures with “oily” odors containing “rummy” and “ethereal” are still predicted accurately. The training and test mixture components have high “oily” content. **e) Orris:** For the “orris” target distribution, 20% also have “spicy” in the test set. Model predictions for “orris” mixtures are poor, with over 40% not assigning any odor label and around 40% predicted as “spicy”. Interestingly, the training and test mixture components have high “orris” content.

The top 5 predicted labels from the odor mix model analysis reveal that “sour”, “onion”, “meaty”, “alliaceous”, and “roasted” perform as expected, with the highest prediction scores for their respective odor labels. Even their mixture components show a high percentage of these odor labels. However, it’s worth noting that odors like “onion”, “alliaceous”, and “roasted” have a second set of odor labels with high co-occurrence in the single molecules dataset. While this strong co-occurrence isn’t observed in mixture labels, the model predicts them with high scores. This suggests the model may struggle to prioritize dominant odors in mixtures, often predicting labels of individual components.

In our analysis, we aimed to address two questions: Firstly, how GNNs capture the relationship between molecular structure and odor; secondly, how GNNs combine their underlying models, captured in the embedding space, for individual molecules into blended pairs. To answer the second question, we analyzed the embedding space.

## References

- [1] Lars Ruddigkeit, Mahendra Awale, and Jean-Louis Reymond. Expanding the fragrance chemical space for virtual screening. *Journal of Cheminformatics*, 6:1–12, 2014.
- [2] Karen J Rossiter. Structure- odor relationships. *Chemical reviews*, 96(8):3201–3240, 1996.
- [3] Andreas Keller, Richard C Gerkin, Yuanfang Guan, Amit Dhurandhar, Gabor Turu, Bence Szalai, Joel D Mainland, Yusuke Ihara, Chung Wen Yu, Russ Wolfinger, et al. Predicting human olfactory perception from chemical features of odor molecules. *Science*, 355(6327):820–826, 2017.
- [4] 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.
- [5] Aaron M Virshup, Julia Contreras-García, Peter Wipf, Weitao Yang, and David N Beratan. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *Journal of the American Chemical Society*, 135(19):7296–7303, 2013.
- [6] RV Krems. Bayesian machine learning for quantum molecular dynamics. *Physical Chemistry Chemical Physics*, 21(25):13392–13410, 2019.
- [7] Sabrina Jaeger, Simone Fulle, and Samo Turk. Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling*, 58(1):27–35, 2018.
- [8] Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.
- [9] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30:595–608, 2016.

- [10] Richard D Hull, Suresh B Singh, Robert B Nachbar, Robert P Sheridan, Simon K Kearsley, and Eugene M Fluder. Latent semantic structure indexing (lassi) for defining chemical similarity. *Journal of medicinal chemistry*, 44(8):1177–1184, 2001.
- [11] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- [12] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.
- [13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [14] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.
- [15] 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, et al. A principal odor map unifies diverse tasks in olfactory perception. *Science*, 381(6661):999–1006, 2023.
- [16] Anne Tromelin, Florian Koensgen, Karine Audouze, Elisabeth Guichard, and Thierry Thomas-Danguin. Exploring the characteristics of an aroma-blending mixture by investigating the network of shared odors and the molecular features of their related odorants. *Molecules*, 25(13):3032, 2020.
- [17] W Luebke. The good scents company information system. *Online Access: http://www. thegoodscentscompany. com*, 2019.
- [18] Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14:217–241, 1994.
- [19] Assaf Kfoury and Laura Sisson. Efficient reassembling of three-regular planar graphs. *Journal of Combinatorial Optimization*, 39(4):1153–1207, 2020.
- [20] John C Leffingwell. Leffingwell & associates, 2005.
- [21] Aryan Amit Barsainyan, Ritesh Kumar, Pinaki Saha, and Michael Schmuker. Multi-labelled smiles odors dataset, 2023.

- [22] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [23] Ritesh Kumar, Rishemjit Kaur, Benjamin Auffarth, and Amol P Bhondekar. Understanding the odour spaces: A step towards solving olfactory stimulus-percept problem. *PLoS One*, 10(10):e0141263, 2015.
- [24] Andrew Dravnieks. Atlas of odor character profiles, 1992.
- [25] E Darío Gutiérrez, Amit Dhurandhar, Andreas Keller, Pablo Meyer, and Guillermo A Cecchi. Predicting natural language descriptions of mono-molecular odorants. *Nature communications*, 9(1):4979, 2018.
- [26] Laura Sisson. Odor descriptor understanding through prompting. *arXiv preprint arXiv:2205.03719*, 2022.
- [27] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [28] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.
- [29] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.
- [30] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- [31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [32] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.