MixScentNet: A Multiscale Graph-based Framework for Predicting Scent Mixture Perception

Xingran Liao[1], **Mingliang Zhou**[2], **Weisi Lin**[1*]
[1]College of Computing and Data Science, Nanyang Technological University
[2]Department of Computer Science, Chongqing University

Abstract

Predicting the olfactory perception of scent mixtures remains a fundamental challenge in computational neuroscience. Existing methods derive mixture-level features by encoding individual components with a single-molecule feature encoder pretrained on annotated olfactory data and aggregating these features via mean pooling, concatenation, or self-attention. This paradigm faces two critical limitations: the scarcity of annotated olfactory data and the inability to capture complex interactions among mixture components. We present MixScentNet, a multiscale graph neural network framework, to address these challenges with two key innovations. First, we introduce a self-supervised pretraining strategy that uses molecular graph structures with RDKit features to predict corresponding Mordred descriptors, enabling the model to learn rich physicochemical knowledge without relying on scarce annotated olfactory data. Second, we propose a novel mixture-as-graph paradigm to model the constituent molecules as nodes in the mixture-level graph, aligning with the fact that humans holistically perceive scent mixtures. We further process the graph via a Graph Attention Network V2 (GATv2) to capture the high-order molecular interactions. MixScentNet differs from existing methods in terms of its pretraining strategy and ability to capture mixture-level features, achieving competitive performance on both mixture-level olfactory label prediction and perceptual distance estimation tasks. We also find that MixScentNet can reproduce olfactory white phenomena, indicating that the model enjoys clear interpretability grounded in psychophysics. The code is available at `https://github.com/Buka-Xing/MixScentNet`

1 Introduction

The human olfactory system is closely related to our experiences of food, beverages, and consumer products such as perfumes, where exposure to pleasant scents can generate considerable commercial value [1, 2]. However, quality control in these industries continues to rely on human sensory panels, which are subjective, costly, and difficult to standardize. The development of computational systems for objective scent prediction is necessary. According to existing research [3, 4, 5], olfactory perception arises from the selective binding of volatile molecules to diverse odourant receptors, which is largely affected by molecular structure and physicochemical properties. With the advent of deep learning, graph neural networks (GNNs) have demonstrated substantial success in predicting diverse chemical properties directly from molecular features [6, 7, 8], establishing a compelling technological basis for modeling the quantitative structure–odor relationship (QSOR).

Current QSOR research focuses primarily on single-molecule olfactory perception, with limited exploration of scent mixtures. Representative works [9, 10, 11] have demonstrated the ability of GNNs to learn complex mappings from molecular features to perceptual descriptors such as "fruity"

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

or "musky". Nevertheless, real-world olfactory stimuli are complex mixtures of tens to hundreds of chemical components. A few approaches [12, 7, 13] explore mixture-level feature encoding by pretraining a single-molecule encoder on annotated olfactory data [14, 15], then independently encoding and aggregating constituent molecule features via attention mechanisms and mean pooling. This paradigm has two fundamental limitations. First, the scarcity of annotated olfactory data critically constrains pretrained feature expressiveness. Even the largest available annotated olfactory database [9] comprises only approximately 8,500 samples, representing sparse coverage of the vast chemical space. Second, as established by Snitz *et al.* [16], human mixture perception is holistic and governed by complex molecular interactions, which can hardly be captured by aggregating independent single-molecule features without modeling their higher-order relationships. These limitations highlight the need for a novel mixture-level framework that jointly addresses data scarcity and high-order molecular interaction modeling.

To address these limitations, we propose MixScentNet with two key designs. First, to mitigate the scarcity of annotated olfactory data, we introduce a self-supervised pretraining strategy for the single-molecule feature extractor. Specifically, the directed message passing neural network (D-MPNN) architecture [17] is adopted as the single-molecule encoder, which is trained with approximately 1 million unlabelled molecules from PubChem [18]. Specifically, molecular graphs are constructed using RDKit [19] node and edge features as inputs, and the D-MPNN [17] is trained to predict the corresponding Mordred descriptors [20], a heterogeneous molecular representation compared to the RDKit features. Such self-supervised training enables the D-MPNN to learn rich physicochemical prior knowledge without the need for perceptual annotations. We also demonstrated that such prior knowledge benefits different QSOR tasks of scent mixture perception. Second, to emulate the holistic nature of human perception and capture high-order molecular interactions, we model each scent mixture as a fully connected graph, where nodes represent the component molecular features. This mixture-level graph is further processed by a graph attention network V2 (GATv2) [21] with 3 layers and residual connections to derive a unified high-order mixture feature. MixScentNet fundamentally differs from prior methods in terms of pretraining the single-molecule feature encoder and deriving mixture-level features, achieving competitive performance in predicting olfactory properties for individual mixtures and estimating the perceptual distance between two mixtures. Moreover, MixScentNet is able to reproduce the established psychophysical phenomenon of *"olfactory white"* [22], demonstrating that it is a psychophysically grounded computational tool for olfaction. Our key contributions are threefold:

- We are the first to introduce self-supervised pretraining to two scent mixture olfactory tasks: olfactory label prediction and perceptual distance estimation. We demonstrate that the olfactory perception of scent mixtures is closely related to the physicochemical properties of the mixture, alleviating the scarcity of annotated data issues in current scent mixture olfactory tasks.
- We design the graph-based modeling of scent mixtures, aligning them with the holistic sensing of human perception. We further employ the GATv2 [21] with 3 layers and residual connections to extract the high-order mixture-level features, aiming to explicitly capture high-order intercomponent relationships that are fundamental to human olfactory perception.
- The results of the experiments reveal that MixScentNet achieves competitive performance across the two olfactory tasks, demonstrating the effectiveness of our designed pipelines. Moreover, MixScentNet is robust to unseen molecules and successfully reproduces the *"olfactory white"* phenomenon, demonstrating the psychophysical validity of our computational framework.

2 Related works

2.1 Perceptual Olfactory Prediction and Distance Estimation

Early works on olfactory perception focused mainly on single-molecule tasks. The POM [10] introduced a principal odor map derived from graph neural network embeddings and demonstrated that a unified molecular feature space can be generalized across diverse olfactory prediction tasks. Mol-PECO [9] proposed a deep position-encoding model that jointly incorporated molecular graph structure and an electronic Coulomb matrix, achieving improved accuracy on the multilabel olfactory annotation prediction task. Recently, the focus has shifted toward the more challenging setting of

scent mixture perception. POMMix [12] used single-molecule features from POM [10] with an attention mechanism to derive mixture-level features, which are then used to analyse the perceptual distance between two scent mixtures. MolSets [7] adopt a similar self-attention mechanism to aggregate mixture-level features from each component feature, which are used to predict lithium battery electrolytes. Sisson *et al.* [13] constructed a large-scale aroma-chemical blend dataset for single molecules and developed OdorPair [13] to predict olfactory labels, which concatenate the paired component molecule graph matrices to form a disconnected mixture graph. SolvGNN [23] constructed a complete molecular interaction graph and applied a one-layer graph convolutional network (GCN) [24] for global feature extraction. Despite these advances, one critical limitation is that the mean pooling, concatenation, or attention mechanisms over single molecular features fail to mimic the holistic sensing mechanism of humans [16]. Moreover, the concatenated graph proposed by OdorPair [13] discards the component identity boundaries and restricts the model to binary mixtures, and the SolvGNN [23] treats all neighbors equally after normalization and cannot be stacked deeper because of oversmoothing [25], which fails to capture high-order molecular interactions. Other methods, such as POMMix [12], rely on annotated single-molecule data for pretraining, leaving the vast space of unlabelled chemical data unexploited. In contrast, MixScentNet employs a self-supervised pretraining strategy over large-scale unlabelled molecular data to learn expressive molecular features and further models the mixture as a complete graph, which is processed by GATv2 [21] to capture high-order molecular interactions.

2.2 GNN for encoding molecular data

The GNNs have become a dominant deep learning tool for encoding molecular features, where three types are commonly employed: the GCN [24], the GAT [26, 21], and the message passing neural network (MPNN) [27]. The three architectures iteratively aggregate information from neighboring nodes to update the hidden features. The key difference lies in how the aggregation is conducted. GCN [24] assigns aggregation weights on the basis of the graph topology and uses symmetric normalization derived from the degree matrix, which treats all neighbors with equal importance. GAT [26] replaces this fixed normalization with a learnable attention mechanism, allowing the model to dynamically weight the contribution of each neighbor. MPNN [27] extends the aggregation by defining explicit message functions over both node and edge features, enabling the model to incorporate rich information during message passing. In our MixScentNet, we adopt two GNN architectures. To encode individual molecular features, we employ the D-MPNN architecture [17] because D-MPNN's bond-centric message passing preserves the direction and chemical semantics of specific bonds, which is critical for capturing single-molecule patterns. The effectiveness of the D-MPNN has also been validated on a wide variety of physicochemical property prediction tasks [28, 17]. To encode the mixture-level graph, we employ GATv2 [21] because GATv2 can dynamically assign different weights to nodes without relying on predefined interaction rules. Moreover, by stacking multiple GATv2 layers, the network progressively refines individual molecular embeddings into high-level semantic features and captures higher-order molecular interactions, which fundamentally differ from the pairwise attention mechanisms between single molecular features employed in POMMix [12] and MolSets [7].

3 Methodology

3.1 Problem formulation and network overview

Denote two scent mixtures as sets of multiple components, $M_1 = \{A_1, A_2, ..., A_n\}$ and $M_2 = \{B_1, B_2, ..., B_m\}$, where $\{A_i | i = 1...n\}$ and $\{B_j | j = 1...m\}$ are the single-molecule components. Our work focuses on two olfactory perception tasks: (1) predicting scores on perceptual descriptors of the given scent mixture M_1 and (2) estimating the perceptual distance between two scent mixtures M_1 and M_2. We address the two tasks with a unified MixScentNet. The mathematical definition of the two problems can be formalized as follows:

$$\begin{cases} \hat{\mathbf{y}}_{M_1} = \text{MLP}[\text{MixScentNet}_\Theta(M_1)] \in \mathbb{R}^K, & \text{Task 1} \\ \hat{d}_{\{M_1,M_2\}} = D(\text{MixScentNet}_\Theta(M_1), \text{MixScentNet}_\Theta(M_2)) \in [0, \infty], & \text{Task 2} \end{cases} \tag{1}$$

where $\hat{\mathbf{y}}_{M_1} \in \mathbb{R}^K$ is the score of the K descriptor. $\text{MixScentNet}_\Theta(\cdot)$ is our proposed MixScentNet with network parameter Θ, which is used to extract mixture-level features of M_1. $\hat{d}_{\{M_1,M_2\}}$

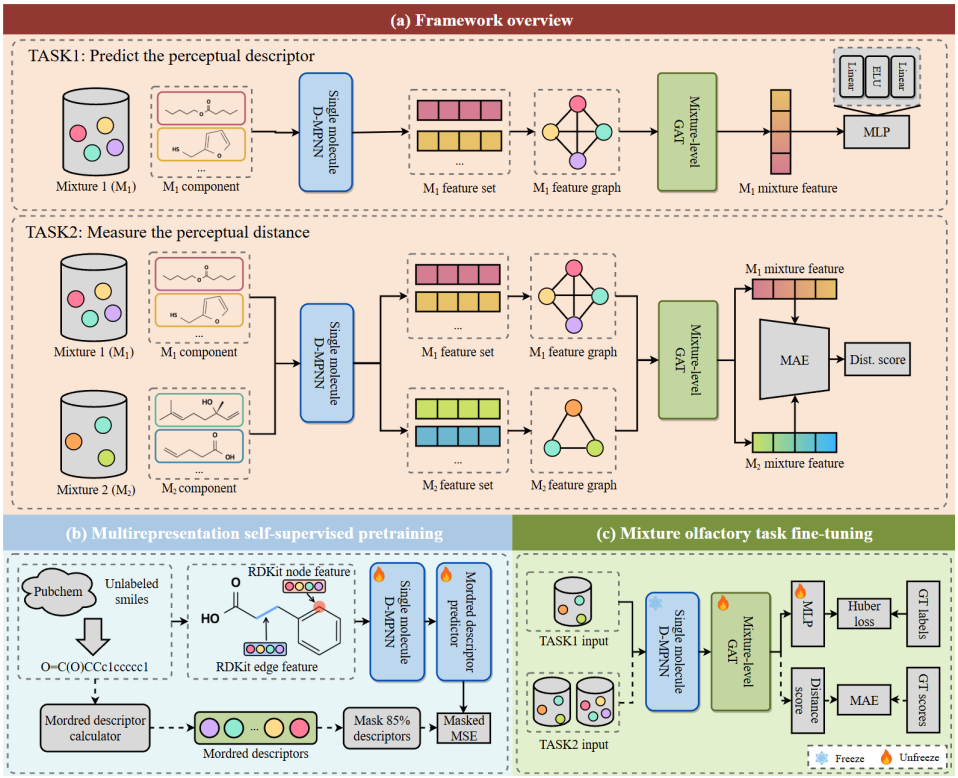

Figure 1: (a) Overview of the proposed framework applied to two olfactory perception tasks. (b) Self-supervised pretraining process for single-molecule feature extraction. (c) Fine-tuning the mixture-level GATv2 [21] on downstream olfactory tasks. The 'MAE' is the mean absolute error.

is the estimated perceptual distance between mixture-level features, and $D(\cdot)$ is a defined mathematical distance metric. We use the mean absolute error (MAE) to measure the feature distance. $\text{MixScentNet}_\Theta(\cdot)$ comprises three main parts: encoding single-molecule features, extracting mixture-level features, and generating task-specific outputs. We depict the overall procedure of the two tasks in Figure 1(a). Specifically, in the inference phase, each input mixture is a set of its component molecules, which are encoded into a set of molecular features via D-MPNN [17]. These individual molecular features are then organized as nodes in a fully connected graph, forming the mixture-level feature graph. GATv2 [21] is used to process this graph to capture the mixture-level feature. For TASK1, the resulting mixture-level feature of M_1 is passed through a multilayer perceptron (MLP) with linear and exponential linear unit (ELU) layers to predict scores on perceptual descriptors. For TASK2, the perceptual distance between M_1 and M_2 is estimated via the MAE of their features.

3.2 Multirepresentation Self-Supervised Pretraining

We introduce a self-supervised pretraining strategy, aiming to address the scarcity of labelled olfactory data and enable the D-MPNN [17] to learn rich physicochemical prior knowledge. The overall process is depicted in Figure 1(b).

Data Preparation and Pretraining Objective The pretraining data are a large number of unlabelled simplified molecular input line entry specification (SMILES) strings sourced from PubChem [18]. Each SMILES string is built into a molecular graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where each node $v \in \mathbb{V}$ and each edge $e \in \mathbb{E}$ are built as the atom-level and the bond-level descriptors derived from the RDKit toolkit [19], respectively. This yields 72 RDKit nodes and 14 bond features in the molecular graph. In parallel, the Mordred [20] descriptor calculator is applied to each SMILES to compute a comprehensive set of physicochemical descriptors, resulting in 1613 total descriptors. Such a descriptor set serves as the pretraining goal. We further perform Z-score normalization to ensure that each descriptor has a zero mean and unit variance and apply winsorization with six standard deviations to replace specific descriptor values outside the assigned range because some Mordred

descriptors have score scales far greater than others. Without processing, these large-scale descriptors will dominate the model's pretraining, preventing the learning of unbiased prior knowledge across all descriptors.

The pretraining task is formulated as a masked descriptor prediction problem [8]. For each molecule, we randomly discard 85% of Mordred descriptors, leaving 242 active descriptors denoted as $\mathbf{d} \in \mathbb{R}^D$. This is directly inspired by ChemBERTa [29]. The model is trained to predict the masked 15% Mordred entries from the RDKit-based molecular graph. Denote $\mathbb{M} \subset \{1, 2, \ldots, 1613\}$ as the set of masked descriptor indices. The D-MPNN $f_\theta(\cdot)$ encodes the molecular graph $\mathbb{G}$ into a molecular embedding $\mathbf{h} \in \mathbb{R}^{2048}$, which is subsequently passed through a lightweight MLP $p_\phi(\cdot)$ to predict the masked Mordred descriptor vector $\hat{\mathbf{d}} = p_\phi(\mathbf{h}) \in \mathbb{R}^D$. The pretraining loss is the mean squared error (MSE) computed over the masked dimensions:

$$\mathcal{L}_{\text{pretrain}} = \frac{1}{|\mathbb{M}|} \sum_{i \in \mathbb{M}} \left(\hat{\mathbf{d}}_i - \mathbf{d}_i \right)^2 . \tag{2}$$

Upon completion of the pretraining, the MLP for Mordred descriptor prediction $p_\phi(\cdot)$ is discarded, and the pretrained D-MPNN $f_\theta(\cdot)$ is carried forward for mixture-level olfactory tasks.

The multirepresentation pretraining strategy offers two key advantages. First, by forcing the D-MPNN [17] to predict a chemically diverse set of Mordred descriptors from RDKit-based graph structures, the model is encouraged to learn a rich knowledge of the physicochemical properties, enabling molecular embeddings to be more informative than those obtained from label-supervised training. Second, the pretraining data include approximately 1 million molecules sampled from PubChem, thereby mitigating the risk of overfitting. In contrast, the data size of annotated olfactory datasets is rather limited, easily leading to overfitting, but constructing a new olfactory database is far more complex.

3.3 Graph-based mixture aggregation

Predicting the olfactory perception of a chemical mixture requires an additional mechanism to capture the holistic, intercomponent relationships within the mixture. We introduce the "mixture-as-a-graph" architecture to model each scent mixture as a fully connected graph and capture the high-order feature interactions by employing GATv2 [21].

Mixture graph construction and feature extraction. Given a chemical mixture M_1 composed of n molecules $\{A_1, A_2, \ldots, A_n\}$, the pretrained D-MPNN [17] $f_\theta(\cdot)$ is first applied to each component independently to obtain a set of molecular embeddings:

$$\mathbf{h}'_i = f_\theta(A_i) \in \mathbb{R}^{2048}, \quad i = 1, 2, \ldots, n. \tag{3}$$

These molecular embeddings are then downsampled by a linear layer $\mathbf{h}_i = \text{Linear}(\mathbf{h}'_i) \in \mathbb{R}^{512}$ and organized as nodes in a fully connected graph $\mathbb{G}_{M_1} = (\mathbb{V}_{M_1}, \mathbb{E}_{M_1})$, where $\mathbb{V}_{M_1} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$ and $\mathbb{E}_{M_1}$ connects all pairs of nodes. This mixture graph $\mathbb{G}_{M_1}$ is subsequently processed by a three-layer GATv2 [21] $g_\Psi(\cdot)$. Specifically, let $\mathbf{H}^{(0)} = \mathbb{V}_{M_1} \in \mathbb{R}^{N \times 512}$ denote the initial node feature matrix. For layers $l = \{1, 2\}$, the node features are updated via multihead attention as follows:

$$\mathbf{H}^{(l)} = \text{LayerNorm}\left(\text{GATv2}^{(l)}\left(\mathbb{G}_{M_1},\, \mathbf{H}^{(l-1)} \right) + \mathbf{W}_{\text{res}}^{(l)} \mathbf{H}^{(l-1)} \right), \tag{4}$$

where $\mathbf{W}_{\text{res}}^{(l)}$ is a linear projection for the residual connection and "LayerNorm" denotes layer normalization. For layers $l \in \{1, 2\}$, the GATv2 [21] operation employs P attention heads and concatenates their outputs, yielding $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times d'}$, where d' is the hidden dimension of the GATv2 [21]. The third layer uses a single attention head to produce the final node representations $\mathbf{H}^{(3)} \in \mathbb{R}^{N \times d'}$, followed by a residual connection and ELU activation:

$$\mathbf{H}^{(3)} = \text{LayerNorm}\left(\text{ELU}\left(\text{GATv2}^{(3)}\left(\mathbb{G}_{M_1},\, \mathbf{H}^{(2)} \right) + \mathbf{H}^{(2)} \right) \right). \tag{5}$$

Finally, we adopt the principal neighborhood aggregation (PNA) [30], which concatenates four statistics over all node features—mean, standard deviation, min, and max—and then project the result via a feed-forward network $\text{FFN}(\cdot)$:

$$\mathbf{z}_{M_1} = g_\Psi(\mathbb{G}_{M_1}) = \text{FFN}\big(\big[\boldsymbol{\mu} \,\|\, \boldsymbol{\sigma} \,\|\, \mathbf{m} \,\|\, \mathbf{M} \big] \big) \in \mathbb{R}^{d'}, \tag{6}$$

where the four statistics are computed as follows:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{H}_i^{(3)}, \quad \boldsymbol{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{H}_i^{(3)}\right)^2 - \boldsymbol{\mu}^2}, \quad \mathbf{m} = \min_i \mathbf{H}_i^{(3)}, \quad \mathbf{M} = \max_i \mathbf{H}_i^{(3)}, \tag{7}$$

with all operations applied elementwise across the N component nodes and $[\,\cdot \,\|\, \cdot\,]$ denoting feature concatenation along the channel dimension. Compared with mean pooling, this PNA preserves both the central tendency ($\boldsymbol{\mu}$) and the dispersion ($\boldsymbol{\sigma}$, $\mathbf{m}$, $\mathbf{M}$) of the per-component embeddings, which is important for olfactory mixtures where a single dominant component or an outlying chemical group can substantially shift the perceived odor. By stacking multiple GATv2 layers with residual connections, the network progressively refines individual molecular embeddings into high-level semantic representations and captures higher-order molecular interactions. Finally, the PNA ensures that the resulting mixture descriptor $\mathbf{z}_{M_1}$ retains the rich distributional structure of those refined node features.

Fine-tuning on downstream tasks. As illustrated in Figure 1(c), the fine-tuning stage freezes the pretrained D-MPNN [17] $f_\theta(\cdot)$ and only trains the GAT $g_\Psi(\cdot)$ along with the MLP heads in TASK1. Specifically, the mixture feature $\mathbf{z}_{M_1}$ is passed through an MLP to predict the perceptual descriptor vector $\hat{\mathbf{y}}_{M_1} \in \mathbb{R}^K$, and the model is optimized via a weighted Huber loss over the K perceptual labels. For each label dimension k, the per-element Huber loss with threshold δ is defined as follows:

$$\ell_\delta(\hat{y}_k, y_k) = \begin{cases} \frac{1}{2}(\hat{y}_k - y_k)^2, & |\hat{y}_k - y_k| < \delta, \\ \delta\left(|\hat{y}_k - y_k| - \frac{1}{2}\delta\right), & \text{otherwise}, \end{cases} \tag{8}$$

where $\delta = 0.4$ is set to accommodate the narrow value range of low-variance perceptual labels while suppressing the influence of outliers via the linear regime, which is selected via hyperparameter searching. The final training objective for TASK1 is as follows:

$$\mathcal{L}_{\text{TASK1}} = \frac{1}{K}\sum_{k=1}^{K}\ell_\delta(\hat{y}_k, y_k), \tag{9}$$

For TASK2, there is no additional output head, and only the GAT $g_\Psi(\cdot)$ is fine-tuned. The mixture-level features $\mathbf{z}_{M_1}$ and $\mathbf{z}_{M_2}$ are independently extracted, and their perceptual distance score is computed as follows:

$$\hat{d}_{\{M_1,M_2\}} = \text{MAE}(\mathbf{z}_{M_1}, \mathbf{z}_{M_2}), \tag{10}$$

where the MAE(*) is the mean absolute error. The fine-tuning objective also minimizes the MAE between the predicted distance scores and the ground-truth perceptual distance scores provided by the human sensory panels:

$$\mathcal{L}_{\text{TASK2}} = \text{MAE}(d_{\{M_1,M_2\}}, \hat{d}_{\{M_1,M_2\}}). \tag{11}$$

4 Experiments

4.1 Experimental setup

Datasets and evaluation settings. MixScentNet is pretrained on the refined PubChem set, whose raw file is collected by Green *et al.* [8], and we detail our refinement process in the Appendix. MixScentNet is evaluated on two olfactory mixture benchmarks: the DREAM Olfactory Mixtures Prediction Challenges 2025 (DREAM2025) [31] and 2024 (DREAM2024) [32]. DREAM2025 [31] contains more than 650 mixtures with 2, 3, 5, or 10 scent molecules. The perceptual labels are collected from trained panelists via the rate-all-that-apply (RATA) method across 51 odor descriptors (*e.g.*, "Cucumber", "Herbal", "Mint"). The objective is to predict the perceptual properties of each mixture as continuous scores over all 51 descriptors. DREAM2024 [32] aims to predict the perceptual distance between two mixtures, which compiles 865 mixture-pair measurements from Snitz *et al.* [16], Ravia *et al.* [33], and Bushdid *et al.* [34]. The scores are unified onto a continuous 0–1 scale, where 0 denotes two indistinguishable mixtures and 1 denotes the maximal perceptual distance. For DREAM2025 [31], following the organizer, we adopt the per-sample average Pearson linear correlation coefficient (PLCC) and cosine distance as evaluation metrics. Specifically, each 51-dimensional predicted score vector is compared with its ground-truth score, and

Table 1: Performance comparison of DREAM2025 (multilabel regression task) and DREAM2024 (perceptual distance prediction task). Bold numbers are the highest mean score in each column. Note that the "Snitz baseline" is only for the perceptual distance task and is not applicable to the label prediction task. "↑" indicates that the larger the score is, the better the performance, and vice versa.

Method (Standard Split)	**DREAM2025 (label)**		**DREAM2024 (distance)**	
	Avg PLCC(↑)	cos-dist (↓)	PLCC (↑)	KRCC (↑)
Snitz Baseline	/	/	0.4154±0.0031	0.3154±0.0532
XGBOOST + MOLT5	0.6726±0.0113	0.2058±0.0060	0.4691±0.0596	0.3669±0.0214
XGBOOST + RDKIT	0.7210±0.0163	0.1745±0.0116	0.4891±0.0644	0.3872±0.0381
XGBOOST + POM	0.6854±0.0145	0.1981±0.0093	0.5243±0.0625	0.4159±0.0125
MolSets	0.5742±0.0212	0.2615±0.0082	0.6175±0.0598	0.4330±0.0490
Chemeleon	0.7466±0.0113	0.1588±0.0056	0.7140±0.0236	0.5163±0.0211
POM+Chemix	0.7277±0.0189	0.1712±0.0133	0.7357±0.0432	0.5569±0.0353
POMMix	0.7496±0.0163	0.1575±0.0104	0.7558±0.0276	**0.5700±0.0258**
MixScentNet	**0.7611±0.0140**	**0.1523±0.0088**	**0.7853±0.0078**	0.5615±0.0126

the overall performance is reported as the average PLCC (Avg PLCC) across all test samples. For DREAM2024 [32], we use the PLCC and the Kendall rank-order correlation coefficient (KRCC) between the predicted and ground-truth perceptual distances. All the experiments are conducted on Ubuntu 16.04 with an Intel(R) Xeon(R) Silver 4214R CPU and an NVIDIA GeForce RTX 3090 GPU. All the models are trained and evaluated via 5-fold cross-validation on both datasets with an 80%/20% training/testing split. The same partitions are shared across all compared models for a fair comparison. Within each fold, the checkpoint with the best PLCC on the test split is taken as the fold result. Such strategy is shared by all compared methods, denoting the upper-bound estimate of the performance of all models. The final performance is reported as the mean plus the standard deviation over the five folds.

Baseline methods and training details. We compare MixScentNet against both classical machine learning baselines and latest deep learning methods. For the classical baselines, we include the Snitz angle metric [16] and XGBoost [35], which are equipped with three different molecular feature sets: MOLT5 [36], RDKit descriptors [19], and POM embeddings [10]. We test these classic methods using the realization proposed by Tom *et al.* [12]. For deep learning comparisons, we evaluate against CheMeleon [8], MolSets [7], and POMMix [12]. CheMeleon [8] is a single-molecule feature encoder that is also pretrained on the PubChem dataset [18], and we use it to generate mixture-level features by summing all the component features. For the label prediction task, the mixture-level feature passes through the same multilabel regression MLP as our MixScentNet. For perceptual distance prediction, the cosine distance is used to compare mixture-level features, yielding the final predicted perceptual distance. During each cross-validation fold, Chemeleon [8] is fine-tuned on the two mixed olfactory datasets. MolSets [7] was originally proposed for predicting the conductivity of lithium battery electrolytes. We adapt it to our tasks by replacing its readout layer with the same multilabel regression head as MixScentNet for DREAM2025 [31] and deriving mixture perceptual distances via the cosine distance between mixture-level features for DREAM2024 [32]. Details of the modification can be found in the Appendix. For POMMix [12], which consists of a pretrained single-molecule feature extractor POM [10] and a mixture-level aggregation module CheMix [12], we evaluate two variants following Tom *et al.* [12]: (i) **POM+CheMix**, where the POM encoder is frozen and only CheMix is fine-tuned on two mixed olfactory datasets; and (ii) **POMMix**, where both the POM encoder and the CheMix module are jointly fine-tuned end-to-end on two mixed datasets.

4.2 Performance Comparison

Table 1 reports the performance of all compared methods on DREAM2025 [31] and DREAM2024 [32]. Among the classical baselines, XGBoost + RDKit achieves competitive label prediction performance (PLCC $= 0.7210$, cos-dist $= 0.1745$), highlighting the critical role of physicochemical properties in olfactory perception [4]. This importance is further evidenced by the gap between CheMeleon [8] and POM+CheMix [12] on DREAM2025. Although CheMeleon is pretrained on unlabelled PubChem [18] data while POM+CheMix uses annotated olfactory data, CheMeleon

Table 2: Performance comparison for unseen molecules split on DREAM2025 (multilabel regression task) and DREAM2024 (perceptual distance prediction task).

Method (Unseen Split)	**DREAM2025 (label)**		**DREAM2024 (distance)**	
	Avg PLCC(↑)	cos-dist (↓)	PLCC (↑)	KRCC (↑)
Snitz Baseline	/	/	0.4262±0.0578	0.3105±0.0610
XGBOOST + MOLT5	0.5756±0.0514	0.2529±0.0310	0.3882±0.1351	0.2322±0.0936
XGBOOST + RDKIT	0.6087±0.0625	0.2338±0.0361	0.4222±0.1074	0.2787±0.0736
XGBOOST + POM	0.6306±0.0260	**0.2237±0.0178**	0.4703±0.0937	0.3280±0.0733
MolSets	0.5360±0.0639	0.2769±0.0332	0.5353±0.0625	0.2769±0.0324
Chemeleon	0.6368±0.0348	0.2239±0.0239	0.5943±0.0719	0.4088±0.0429
POM+Chemix	0.6236±0.0374	0.2393±0.0272	0.6124±0.0906	0.4202±0.0571
POMMix	0.6354±0.0458	0.2336±0.0314	0.6015±0.1039	0.4233±0.0550
MixScentNet	**0.6495±0.0364**	0.2262±0.0265	**0.6724±0.0842**	**0.4318±0.0543**

outperforms POM+CheMix (PLCC 0.7466 vs. 0.7277) and remains competitive with POMMix [12], confirming that self-supervised pretraining on large-scale unlabelled chemical data provides richer prior knowledge that is effective in scent mixture perception tasks. However, physicochemical priors alone are insufficient for perceptual distance estimation. On DREAM2024, CheMeleon becomes inferior to POMMix (PLCC 0.7140 vs. 0.7558, KRCC 0.5163 vs. 0.5700). We attribute this to two factors: (i) we apply naive summation aggregation on CheMeleon features for distance estimation, which ignores molecular interactions; and (ii) POMMix employs a scaled cosine distance regression head [12] that recalibrates raw cosine distances to handle the noisy label distribution in DREAM2024, where identical mixture pairs are often assigned nonzero perceptual distances because of human perceptual noise. In summary, capturing rich physicochemical priors and modeling molecular interactions are both indispensable. MixScentNet addresses both and outperforms all the baselines on DREAM2025 (PLCC 0.7611, cos-dist 0.1523) while achieving the best PLCC (0.7853) on DREAM2024. These gains stem from two innovations: self-supervised pretraining on $\sim$1 M unlabelled SMILES strings from PubChem [18], which yields richer physicochemical priors than label-supervised pretraining does; and the mixture-as-graph paradigm with a three-layer GATv2, which progressively captures higher-order molecular interactions reflecting the holistic nature of human olfactory mixture perception [16].

4.3 Model robustness to unseen molecules

We further evaluate the generalization capability of MixScentNet under an **unseen-molecule** cross-validation protocol. For each fold, we first collect the unique molecular SMILES list across the entire dataset and randomly shuffle it. We then iterate over the shuffled list, marking each molecule as **unseen** and assigning any sample containing it to the test set until the test set reaches approximately 20% of the total samples and assign the remaining 80% to the training set. All the models share the same training/testing splits per fold, and the results are reported in Table 2. As expected, all methods exhibit a notable performance decrease relative to the standard cross-validation results in Table 1. Nevertheless, MixScentNet achieves robust performance across both tasks (PLCC $= 0.6495$, cos-dist $= 0.2262$ on DREAM2025 [31]; PLCC $= 0.6724$, KRCC $= 0.4318$ on DREAM2024 [32]), with a notable 11.79% PLCC improvement over POMMix [12] on DREAM2024. We attribute this robustness to two design choices. First, MixScentNet acquires rich physicochemical priors from large-scale self-supervised pretraining. Although POMMix also uses pretraining, its $\sim$5K-annotated olfactory samples cover only a sparse region of chemical space, leading to inferior generalization on DREAM2024 (PLCC 0.6015 vs. 0.6724). Second, our mixture-level graph encoding captures higher-order molecular interactions that are critical for generalization. This is evidenced by CheMeleon [8], which uses the same self-supervised pretraining but lacks explicit interaction modeling. Finally, Chemeleon [8] falls behind MixScentNet on DREAM2024 (PLCC 0.5943 vs. 0.6724). The cos-dist of MixScentNet on DREAM2025, while not the best, closely matches the top-performing XGBoost + POM (0.2262 vs. 0.2237). We attribute this to training MixScentNet with MAE loss, which focuses on minimizing absolute score differences rather than preserving overall score-distribution similarity.

Table 3: Comparison with official submission leaderboards on DREAM2025 (perceptual label prediction) and DREAM2024 (perceptual distance prediction).

Method	Avg PLCC (↑)	cos-dist (↓)	Rank (↓)
PL21	0.8043	0.1280	1
PL21	0.8053	0.1283	1
@yuanfang.guan	0.7928	0.1372	3
IM-Scent	0.7798	0.1448	4
@yuanfang.guan	0.7712	0.1477	5
PL21	0.7741	0.1495	5
Nylix	0.7664	0.1530	7
@yuanfang.guan	0.7554	0.1578	8
@jgburk	0.7394	0.1690	9
IM-Scent	0.7268	0.1825	10
MixScentNet	0.7885	0.1378	3.5

(a) DREAM2025 (label prediction)

Method	MSE (↓)	MSE Rank (↓)	PLCC (↑)	PLCC Rank (↓)	Avg Rank (↓)
UMich CASI (1st)	0.007874	3	0.5011	2	2.5
ChemSenSim Lab (1st)	0.008578	5	0.5012	1	3
belfaction (1st)	0.007810	2	0.4302	5	3.5
D2Smell (1st)	0.0062	1	0.6075	1	1
ruhallah1020	0.009114	7	0.4668	3	5
PL21	0.007525	1	0.3581	10	5.5
Olfactory24UNIBA	0.008501	4	0.4038	7	5.5
Tamarin	0.008868	6	0.4032	8	7
who_nose	0.013576	12	0.4549	4	8
Songxuebo	0.013520	11	0.3725	9	10
MixScentNet	0.0089	6.5	0.5316	1.5	4

(b) DREAM2024 (perceptual distance estimation)

4.4 Performance on the official splits

We further compare MixScentNet against top leaderboard submissions on DREAM2024 and DREAM2025 using official train-validation-test splits. For DREAM2025 [31], we train on the official training set and evaluate on the validation set, comparing against the top-10 teams on the validation leaderboard[2] (Table 3a). For DREAM2024 [32], we use the official split from the organiser's GitHub repository[3] and compare against the top-10 teams on the official test leaderboard[4] (Table 3b). The DREAM2024 validation leaderboard is excluded due to severe data leakage during the competition phase[5]. On DREAM2025, MixScentNet achieves a Pearson correlation of 0.7885 and cosine distance of 0.1378, placing approximately 3rd between @yuanfang.guan (0.7928, rank 3) and IM-Scent (0.7798, rank 4). On DREAM2024, MixScentNet achieves a PLCC of 0.5316 (rank 1.5th), surpassing three of the four first-place teams, including UMich CASI (0.501), ChemSenSim Lab (0.501), and belfaction (0.430). Only D2Smell (0.607) performs better. The weaker MSE ranking (6.5th) reflects that MixScentNet is trained with MAE rather than MSE loss. We attribute such competitiveness to two biologically grounded design choices. First, self-supervised pre-training on molecular physicochemical descriptors. Olfactory perception is governed by the selective binding of scent molecules to ORs, a process largely determined by molecular physicochemical properties [37]. Our work demonstrates that these priors transfer effectively to mixture perception tasks, which is non-trivial as even the latest DREAM submissions still rely on annotated mono-molecular datasets [10] with limited data for pre-training. Second, the mixture-as-graph design: Xu et al. [37] showed that odor mixture perception involves inherently non-additive receptor-level modulation, such as agonism, antagonism, and mixture-emergent activation. They proposed a modulation model that treats the mixture as a holistic entity. Our mixture-as-graph directly realizes this model by representing the mixture as a unified graph in which each molecular node's embedding is dynamically modulated by all other nodes through GATv2 attention, capturing higher-order intermolecular interactions.

4.5 Reproducing the olfactory white phenomena

The *olfactory white* phenomenon [22] describes the perceptual convergence of sufficiently complex odor mixtures: as the number of diverse components increases, the resulting mixture perception loses its distinctive character and converges toward a neutral odor, regardless of molecular identity. This is analogous to white noise in audition or white light in vision. To examine whether MixScentNet captures this property, we conduct a leave-one-subset-out evaluation on DREAM2024 [32] (Snitz [16], Ravia [33], Bushdid [34]), training on two subsets and testing on the held-out one. The representative result is shown in Figure 2 (trained on Snitz+Bushdid, tested on Ravia). Each point represents a mixture pair with horizontal coordinate $\sqrt{n_1 \cdot n_2}$ and vertical coordinate the perceptual distance. Black error bars indicate the within-bin mean $\pm$ standard deviation, and the red line is an OLS regression over all points. The significant negative slope confirms that predicted distances decrease as mixture complexity grows. Moreover, MixScentNet shows a stronger negative correlation than ground-truth human judgments (Pearson: -0.2358 vs. -0.1190; Spearman: -0.2866 vs. -0.2090), suggesting the model captures perceptual convergence more consistently than crowdsourced ratings

[2] https://www.synapse.org/Synapse:syn53470621/wiki/627131

[3] https://github.com/Satarifard/DREAM-olfactory-mixtures-prediction-challenge/

[4] https://www.synapse.org/Synapse:syn66484079/tables/

[5] https://www.synapse.org/Synapse:syn53470621/discussion/threadId=11312

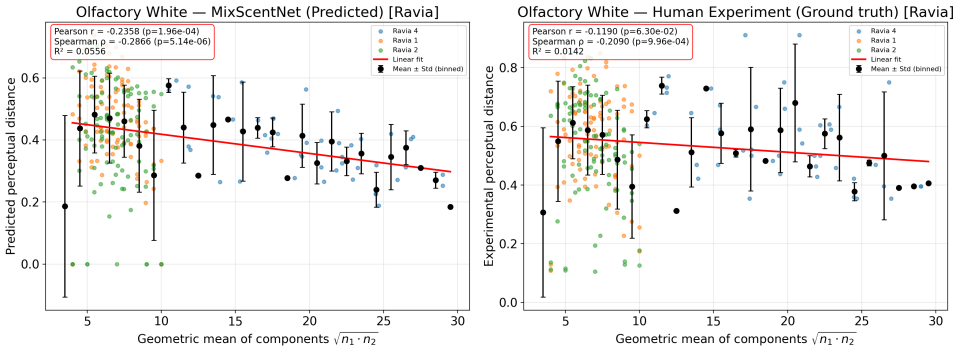

Figure 2: Reproducing the *olfactory white* phenomena on the Ravia [33] subsets in DREAM2024 [32], where MixScentNet is trained on Snitz [16] and Bushdid [34].

subject to individual noise. Additional results on other splits are provided in Appendix A.2, delivering the same trends. In conclusion, MixScentNet exhibits a perceptual structure closely aligned with the human olfactory system.

5 Limitations and potential social impact

MixScentNet has two limitations. First, both DREAM2025 [31] and DREAM2024 [32] contain limited annotated mixtures, and currently, there are no large-scale mixed olfactory datasets that enable more conclusive assessments. Second, our model relies mainly on molecular structure and does not incorporate concentration, intensity, or temporal dynamics, all of which are known to modulate human odor perception. Extending the framework to concentration-aware or time-resolved settings is our future work. As for the social impact, MixScentNet is unlikely to be used for harmful purposes because of the benign nature of olfactory prediction. MixScentNet may also benefit several downstream applications by reducing the reliance on costly and time-consuming human panel evaluations, including fragrance and flavour design, beverage formulation, and environmental odor monitoring. We also hope that MixScentNet can serve as an extensible foundation for future research in computational olfaction.

6 Conclusion

We present MixScentNet, a graph neural network framework for olfactory perception prediction of molecular mixtures, addressing two critical limitations of existing methods: the scarcity of labeled olfactory data and the inability to model high-order molecular interactions. MixScentNet introduces self-supervised pretraining on large-scale unlabelled SMILES strings from PubChem [18] and a mixture-as-graph paradigm processed by a three-layer GATv2 [21] with residual connections. Experiments on DREAM2025 [31] and DREAM2024 [32] demonstrate the competitive performance of MixScentNet on both tasks and strong generalization under the unseen-molecule cross-validation protocol. MixScentNet further reproduces established psychophysical phenomena, the *olfactory white* [22], providing scientific grounding beyond benchmark metrics. We hope that MixScentNet can serve as an extensible foundation for future research in computational olfaction, with applications spanning fragrance design and food science.

Funding Statement & Competing Interests

This work was supported in part by the Agency for Science, Technology, and Research (A*STAR) under its MTC Programmatic Funding Scheme (project M23L8b0049) Scent Digitalization and Computation (SDC) Program, and partially supported by the Ministry of Education, Singapore, under the funding of Tier 1, RG103/24, and partially by the National Science Foundation of China under Grant 62672074. The authors declare no competing interests.

References

[1] Michiel Schreurs, Supinya Piampongsant, Miguel Roncoroni, Lloyd Cool, Beatriz Herrera-Malaver, Christophe Vanderaa, Florian A Theßeling, Łukasz Kreft, Alexander Botzki, Philippe Malcorps, et al. Predicting and improving complex beer flavor through machine learning. *Nature Communications*, 15(1):2368, 2024.

[2] Chen Chen, Meiqi Sheng, Haibin Yuan, Xin Pan, Huaixiang Tian, and Xinman Lou. Perceptual interaction of aroma compounds and their effects on the enhancement of flavor quality in foods: A review. *Trends in Food Science & Technology*, page 105315, 2025.

[3] Heinz Breer, Ina Wanner, and Jörg Strotmann. Molecular genetics of mammalian olfaction. *Behavior genetics*, 26(3):209–219, 1996.

[4] Anju Sharma, Rajnish Kumar, Imlimaong Aier, Rahul Semwal, Pankaj Tyagi, and Pritish Varadwaj. Sense of smell: structural, functional, mechanistic advancements and challenges in human olfactory research. *Current neuropharmacology*, 17(9):891–911, 2019.

[5] Rayane Achebouche, Anne Tromelin, Karine Audouze, and Olivier Taboureau. Application of artificial intelligence to decode the relationships between smell, olfactory receptors and small molecules. *Scientific reports*, 12(1):18817, 2022.

[6] Mehmet Muezzinoglu, Alexander Vergara, Ramon Huerta, Thomas Nowotny, Nikolai Rulkov, Henry Abarbanel, Allen Selverston, and Mikhail Rabinovich. Artificial olfactory brain for mixture identification. *Advances in Neural Information Processing Systems*, 21, 2008.

[7] Hengrui Zhang, Tianxing Lai, Jie Chen, Arumugam Manthiram, James M. Rondinelli, and Wei Chen. Learning molecular mixture property using chemistry-aware graph neural network. *PRX Energy*, 3:023006, Jun 2024.

[8] William Green, Jackson Burns, Akshat Shirish Zalte, Charlles Abreu, Jochen Sieg, Christian Feldmann, and Miriam Mathea. Deep learning foundation models from classical molecular descriptors. 2026.

[9] Mengji Zhang, Yusuke Hiki, Akira Funahashi, and Tetsuya J Kobayashi. A deep position-encoding model for predicting olfactory perception from molecular structures and electrostatics. *npj Systems Biology and Applications*, 10(1):76, 2024.

[10] 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.

[11] Yu Wang, Qilong Zhao, Mingyuan Ma, and Jin Xu. Olfactory perception prediction model inspired by olfactory lateral inhibition and deep feature combination. *Applied Intelligence*, 53(16):19672–19684, 2023.

[12] Gary Tom, Cher Tian Ser, Ella M Rajaonson, Stanley Lo, Hyun Suk Park, Brian K Lee, and Benjamin Sanchez-Lengeling. Does this smell the same? learning representations of olfactory mixtures using inductive biases. *Machine Learning: Science and Technology*, 6(3):035063, 2025.

[13] Laura Sisson, Aryan Amit Barsainyan, Mrityunjay Sharma, and Ritesh Kumar. Deep learning for odor prediction on aroma-chemical blends. *ACS omega*, 10(9):8980–8992, 2025.

[14] The Good Scents Company. The good scents company information system. `https://www.thegoodscentscompany.com/`. Accessed: 2026-04-11.

[15] John C. Leffingwell. PMP 2001 – Perfume Materials Performance Database (BACIS). `http://www.leffingwell.com/bacispmp.htm`. Leffingwell & Associates. Accessed: 2026-04-11.

[16] Kobi Snitz, Adi Yablonka, Tali Weiss, Idan Frumin, Rehan M Khan, and Noam Sobel. Predicting odor perceptual similarity from odor structure. *PLoS computational biology*, 9(9):e1003184, 2013.

[17] Esther Heid, Kevin P Greenman, Yunsie Chung, Shih-Cheng Li, David E Graff, Florence H Vermeire, Haoyang Wu, William H Green, and Charles J McGill. Chemprop: a machine learning package for chemical property prediction. *Journal of chemical information and modeling*, 64(1):9–17, 2023.

[18] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 2023.

[19] RDKit Contributors. Rdkit: Open-source cheminformatics. `https://www.rdkit.org`, 2006–. Accessed: 2026-04-08.

[20] Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of cheminformatics*, 10(1):4, 2018.

[21] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[22] Tali Weiss, Kobi Snitz, Adi Yablonka, Rehan M Khan, Danyel Gafsou, Elad Schneidman, and Noam Sobel. Perceptual convergence of multi-component mixtures in olfaction implies an olfactory white. *Proceedings of the National Academy of Sciences*, 109(49):19959–19964, 2012.

[23] Shiyi Qin, Shengli Jiang, Jianping Li, Prasanna Balaprakash, Reid C Van Lehn, and Victor M Zavala. Capturing molecular interactions in graph neural networks: a case study in multi-component phase equilibrium. *Digital Discovery*, 2(1):138–151, 2023.

[24] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

[25] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. accepted as poster.

[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] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[29] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[30] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in neural information processing systems*, 33:13260–13271, 2020.

[31] Sage Bionetworks. DREAM olfactory mixtures prediction challenge 2025. `https://www.synapse.org/Synapse:syn64743570/wiki/632379`, 2025. Synapse ID: syn64743570. Accessed: 2026-04-16.

[32] Vahid Satarifard, Laura Sisson, Yikun Han, Pedro Ilídio, Matej Hladiš, Maxence Lalis, Xuebo Song, Wenjie Yin, Aharon Ravia, CiCi Xingyu Zheng, et al. High-fidelity tuning of olfactory mixture distances in the perceptual space of smell through a community effort. *biorxiv*, pages 2025–12, 2025.

[33] Aharon Ravia, Kobi Snitz, Danielle Honigstein, Maya Finkel, Rotem Zirler, Ofer Perl, Lavi Secundo, Christophe Laudamiel, David Harel, and Noam Sobel. A measure of smell enables the creation of olfactory metamers. *Nature*, 588(7836):118–123, 2020.

[34] Caroline Bushdid, Marcelo O Magnasco, Leslie B Vosshall, and Andreas Keller. Humans can discriminate more than 1 trillion olfactory stimuli. *Science*, 343(6177):1370–1372, 2014.

[35] Pankaj Tyagi, Anju Sharma, Rahul Semwal, Uma Shanker Tiwary, and Pritish Kumar Varadwaj. Xgboost odor prediction model: finding the structure-odor relationship of odorant molecules using the extreme gradient boosting algorithm. *Journal of Biomolecular Structure and Dynamics*, 42(20):10727–10738, 2024.

[36] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, 2022.

[37] Lu Xu, Wenze Li, Venkatakaushik Voleti, Dong-Jing Zou, Elizabeth MC Hillman, and Stuart Firestein. Widespread receptor-driven modulation in peripheral olfactory coding. *Science*, 368(6487):eaaz5390, 2020.

[38] Emily J Mayhew, Charles J Arayata, Richard C Gerkin, Brian K Lee, Jonathan M Magill, Lindsey L Snyder, Kelsie A Little, Chung Wen Yu, and Joel D Mainland. Transport features predict if a molecule is odorous. *Proceedings of the National Academy of Sciences*, 119(15):e2116576119, 2022.

[39] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Table A.1: Ablation studies on single-molecule encoders and the mixture graph network.

Method	DREAM2025 (label)		DREAM2024 (distance)	
	Avg PLCC(↑)	cos-dist (↓)	PLCC (↑)	KRCC (↑)
POM(GS-LF)+GATv2	0.7245±0.0188	0.1752±0.0108	0.7497±0.0135	0.5481±0.0155
POM(PubChem)+GATv2	0.7348±0.0138	0.1741±0.0091	0.7689±0.0082	0.5563±0.0128
D-MPNN(GS-LF)+GATv2	0.7328±0.0215	0.1768±0.0129	0.7515±0.0186	0.5489±0.0203
MixScentNet *w/* 1 layer-GCN	0.7568±0.0156	0.1647±0.0079	0.7618±0.0106	0.5511±0.0132
MixScentNet *w/* 2 layer-GCN	0.7478±0.0149	0.1713±0.0115	0.7559±0.0131	0.5467±0.0148
MixScentNet *w/* 3 layer-GCN	0.7405±0.0153	0.1769±0.0106	0.7438±0.0117	0.5489±0.0135
MixScentNet	**0.7611±0.0140**	**0.1523±0.0088**	**0.7853±0.0078**	**0.5615±0.0126**

A Appendix

In the Appendix, we first present ablation studies in Appendix A.1 and additional olfactory white results in Appendix A.2. To ensure reproducibility, we further specify the details in Appendix A.3, A.4, and A.5, including the datasets used, MixScentNet implementation and training procedures, and baseline method configurations.

A.1 Ablation studies

Different single-molecule encoders. We first investigate the effect of the single-molecule encoder by introducing three variants, each pairing a different pretrained encoder on specific datasets with the 3-layer residual GATv2 [21]: "POM(GS-LF)+GATv2", "POM(PubChem)+GATv2", and "D-MPNN(GS-LF)+GATv2". The GS-LF [10] dataset contains 4,983 molecular SMILES strings and is annotated with single-molecule olfactory labels. For the GS-LF set [10], the encoders are pretrained on the full set using binary cross-entropy loss and the Adam optimizer. We select the epoch with the highest AUROC as the final checkpoint. For PubChem [18], the data preparation and training details are described in the Appendix A.4. The single-molecule encoder is then **frozen**, and only GATv2 [21] is fine-tuned on DREAM2025 [31] and DREAM2024 [32] under the 5-fold 80%/20% train-test split.

We report the mean and standard deviation across folds in Table A.1, where the performance gaps reveal three valuable insights. First, the scale of the pretraining datasets significantly influences the model performance. Although the GS-LF [10] set contains annotated molecular data, both "POM(GS-LF)+GATv2" (PLCC $= 0.7245$) and "D-MPNN(GS-LF)+GATv2" (PLCC $= 0.7328$) are inferior to the "POM(PubChem)+GATv2" (PLCC $= 0.7348$) on DREAM2025 [31], and this gap even widens on DREAM2024 [32] ("POM(GS-LF)+GATv2" PLCC$= 0.7497$ and "D-MPNN(GS-LF)+GATv2" PLCC$= 0.7515$ vs. "POM(PubChem)+GATv2" PLCC$= 0.7689$). This finding indicates that models pretrained on a small task-aligned dataset may fail to generalize. Second, the encoder architecture has a marginal effect when training on the small-scale GS-LF set [10]. We find that "POM(GS-LF)+GATv2" and "D-MPNN(GS-LF)+GATv2" differ by less than 0.01 in the PLCC on both tasks, suggesting that the prior knowledge is fully learned because of the limited dataset scale. Third, MixScentNet outperforms all three variants on every metric (PLCC $= 0.7611$ and cos-dist $= 0.1523$ on DREAM2025; PLCC $= 0.7853$ and KRCC $= 0.5615$ on DREAM2024), improving over the strongest variant "POM(PubChem)+GATv2" by $+2.63\%$ PLCC on DREAM2025 and $+1.64\%$ PLCC on DREAM2024. We believe this improvement is because the D-MPNN [17] enjoys a larger model capacity than the POM [10], avoiding the underfitting issue on the large-scale PubChem [18] dataset. In conclusion, self-supervised pretraining with Mordred descriptors outperforms GS-LF supervised pretraining, suggesting that physicochemical priors may transfer effectively to mixture-level olfactory tasks. Whether this advantage generalizes to other self-supervised objectives, such as the Morgan fingerprint, remains an open question for future work.

GCN [24] vs. GATv2 [21] in processing the mixture-level graph. In SolvGNN [23], Qin *et al.* employ a GCN [24] to process mixture-level graphs containing only 2–3 components. However, in our olfactory tasks, where mixtures frequently contain more than 10 components, we find that the GCN [24] is not the best choice for processing fully connected mixture-level graphs. We show ablation studies in Table A.1, where we use the GCN [24] to process the mixture-level graph and

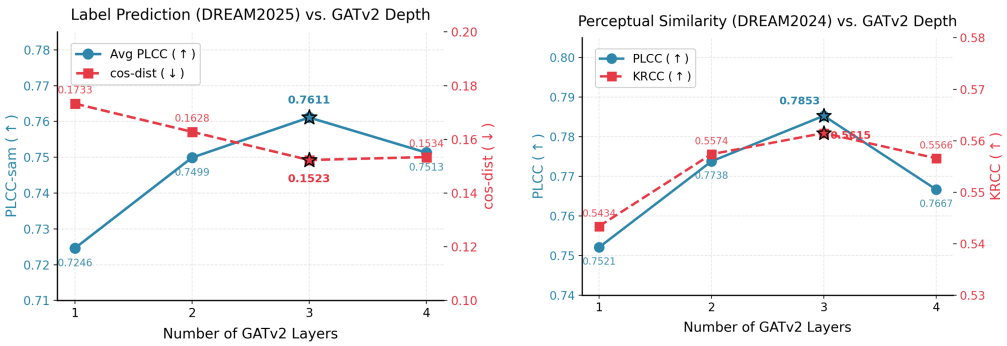

Figure A.1: Ablation studies on the depth of GAT layers.

explore the influence of stacking more GCN layers. On DREAM2025 [31], the single-layer GCN variant ("MixScentNet w/1 layer-GCN") reaches a PLCC of $= 0.7568$, which is marginally below that of MixScentNet (PLCC $= 0.7611$). However, on DREAM2024 [32], the performance of the single-layer GCN variant is largely inferior to that of the default MixScentNet (PLCC 0.7618 vs. 0.7853). Moreover, the performance decreases when more GCN layers are stacked. The PLCC decreases to 0.7478 with two layers and 0.7405 with three layers on DREAM2025 [31]. The DREAM2024 [32] results show the same trend: the PLCC decreases from 0.7618 (1 layer) to 0.7559 (2 layers) and 0.7438 (3 layers), a -1.80% decrease compared with our 3-layer GATv2 design (PLCC $= 0.7853$). We believe that this degradation is the signature of oversmoothing [25]. Specifically, in a fully connected mixture graph with $n \geq 10$ nodes, every node receives a uniformly weighted aggregate of all the others at every layer, and each component embedding collapses toward a common mean after multiple GCN layers, after which the chemical identities that are essential in the downstream tasks are removed. GATv2 [21] avoids this failure through learnable attention weights to suppress uninformative neighbors and aggregate information selectively, allowing more layers to be stacked to refine representations rather than dilute information. Therefore, our proposed MixScentNet outperforms the best GCN variant by $+0.43\%$ PLCC on DREAM2025 and $+2.35\%$ PLCC on DREAM2024.

Layers of the GATv2 network. We explore how the depth of the residual GATv2 [21] stack affects the performance on the two olfactory tasks. We stack GATv2 layers from 1 to 4 while keeping all the other components unchanged and report the results in Figure A.1(a, b). On both downstream tasks, the performance improves substantially as the depth increases from the 1 layer to the 3 layer and then decreases when a fourth layer is added. In Figure A.1(a), the Avg PLCC increases from 0.7246 at the 1 layer to a peak of 0.7611 at the 3 layer (a relative gain of $+5.04\%$) and decreases to 0.7513 at the 4 layer. In Figure A.1(b), the PLCC and KRCC show similar variation trends, increasing from $0.7521/0.5434$ at the 1 layer to $0.7853/0.5615$ at the 3 layer and then decreasing to $0.7667/0.5566$ at the 4 layer. We attribute these observations to the increasing expressiveness of higher-order interactions. Specifically, a single GATv2 layer can capture only pairwise dependencies between molecules, whereas stacking three layers allows each component to be repeatedly contextualized against all other components, refining the representation through multihop attention. However, stacking more layers does not consistently enhance the performance. The degradation is shown at the 4 layers, indicating the overfitting issue on the two small-scale mixed olfactory datasets. In conclusion, we choose 3 layers to pursue the best trade-off as the default configuration of MixScentNet.

Different pooling strategies for the GATv2. To aggregate the final mixture-level feature, we compare two strategies: mean pooling (MeanPool) over all refined node embeddings and default *PNA* [30]. The results are shown in Figure A.2(a). We find that MeanPool slightly outperforms PNA, but the margins are below 0.4% on DREAM2025 [31], whereas PNA [30] largely outperforms MeanPool (PLCC 0.7853 v.s. 0.7645; KRCC 0.5615 v.s. 0.5405) on DREAM2024 [32]. We suspect that this is because the perceptual distance estimation involves comparing two mixtures with different molecular compositions and varying component counts, and the distributional structure of the node embeddings is more discriminative than that of only MeanPool. Specifically, two mixtures may share similar mean embeddings but differ markedly in their dispersion or extremes; the PNA preserves these differences and allows the downstream distance head to exploit them. In contrast, for the

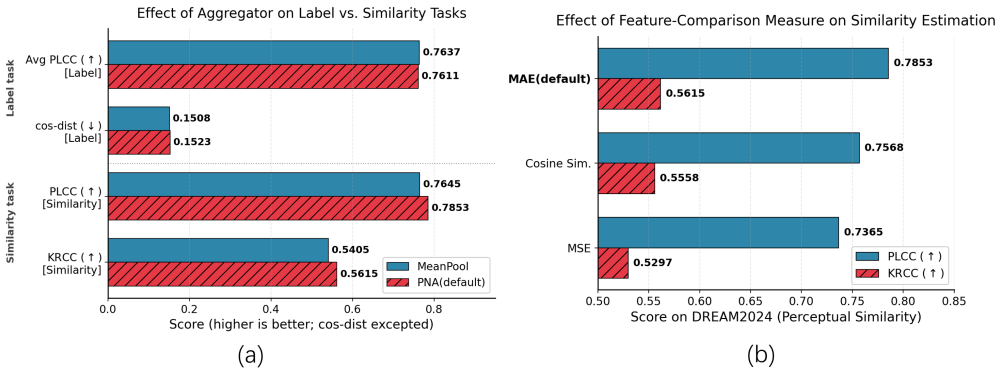

Figure A.2: Ablation studies on the pooling strategies and the feature comparison method.

label prediction task in DREAM2025 [31], the 51 perceptual descriptors are essentially *averaged* over panelists and over components, and a mean readout naturally matches this averaging structure. Considering the overall performance, we select the PNA [30] as the default setting of MixScentNet.

Different feature-comparison measures. To estimate the perceptual distance, how to compare two mixture embeddings is a critical design decision. We compare three alternatives, namely, the MAE, cosine distance, and mean squared error (MSE), on DREAM2024 [32]. The results are depicted in Figure A.2(b). The MAE-based measure achieves the best performance on both metrics (PLCC 0.7853; KRCC 0.5615), outperforming the cosine similarity (PLCC 0.7568; KRCC 0.5558) and the MSE (PLCC 0.7365 0.5297) by $+6.62\%$ PLCC. We suspect that the gap between the MAE and cosine similarity is because the cosine similarity discards the magnitude information of the embeddings and measures only their angular alignment. Nevertheless, the componentwise magnitudes may carry a meaningful signal in measuring the difference between mixture embeddings. On the other hand, the MSE penalizes large per-dimension residuals quadratically, which may overweight a few outlier dimensions at the expense of overall calibration. In contrast, the MAE leads to a more uniform gradient signal, aligning better with the human-rated perceptual distances. In conclusion, we believe that preserving magnitude while remaining robust to outlier dimensions explains why the MAE is the most appropriate choice for the perceptual distance objective.

A.2 More "*olfactory white*" reproducing results

We present two additional *olfactory white* reproduction results on DREAM2024 [32] in Figure A.3. For each configuration, we plot the predicted perceptual distance against the geometric mean of the component counts $\sqrt{n_1 \cdot n_2}$ for every mixture pair. The error bars are the binned means and standard deviations. The red line is the OLS linear fit computed over all the raw points. For "[Ravia, Bushdid] $\rightarrow$ Snitz" (upper of Figure A.3), we find that both the model and the human ground truth exhibit strong, significant negative correlations. MixScentNet yields Pearson $r = -0.2896$ with $p = 2.19 \times 10^{-8}$ and Spearman $\rho = -0.1670$ with $p = 1.48 \times 10^{-3}$, successfully reproducing the "*olfactory white*" phenomena. More interestingly, for "[Snitz, Ravia] $\rightarrow$ Bushdid" (lower of Figure A.3), we find that MixScentNet produces a statistically significant negative trend, with Pearson $r = -0.2597$ ($p = 2.23 \times 10^{-5}$) and Spearman $\rho = -0.2595$ ($p = 2.27 \times 10^{-5}$). Strikingly, the human ground-truth perceptual distances on the same Bushdid pairs exhibit nearly no correlation with $\sqrt{n_1 \cdot n_2}$ (Pearson $r = -0.0546$, $p = 0.38$; Spearman $\rho = -0.0532$, $p = 0.39$), and the trend is not statistically significant. We suspect that the reason is that the Bushdid subset was collected via a triangle test paradigm [34], whose discrimination success rate measure is noisier at the individual-pair level than direct similarity ratings are, so the underlying *olfactory white* trend can be masked by human perceptual noise. Nevertheless, our proposed MixScentNet can still recover the expected negative trend more clearly than human panels can, indicating that it enjoys robustness to perceptual noise. Most importantly, MixScentNet produces a statistically significant negative trend ($p < 10^{-3}$) in every split, demonstrating that it can reproduce the *olfactory white* phenomenon regardless of the train–test split choice. This behavior emerges without any explicit olfactory-white supervision but arises purely from training on pairwise similarity labels, which supports the claim that the MixScentNet aligns with human perception at a structural level.

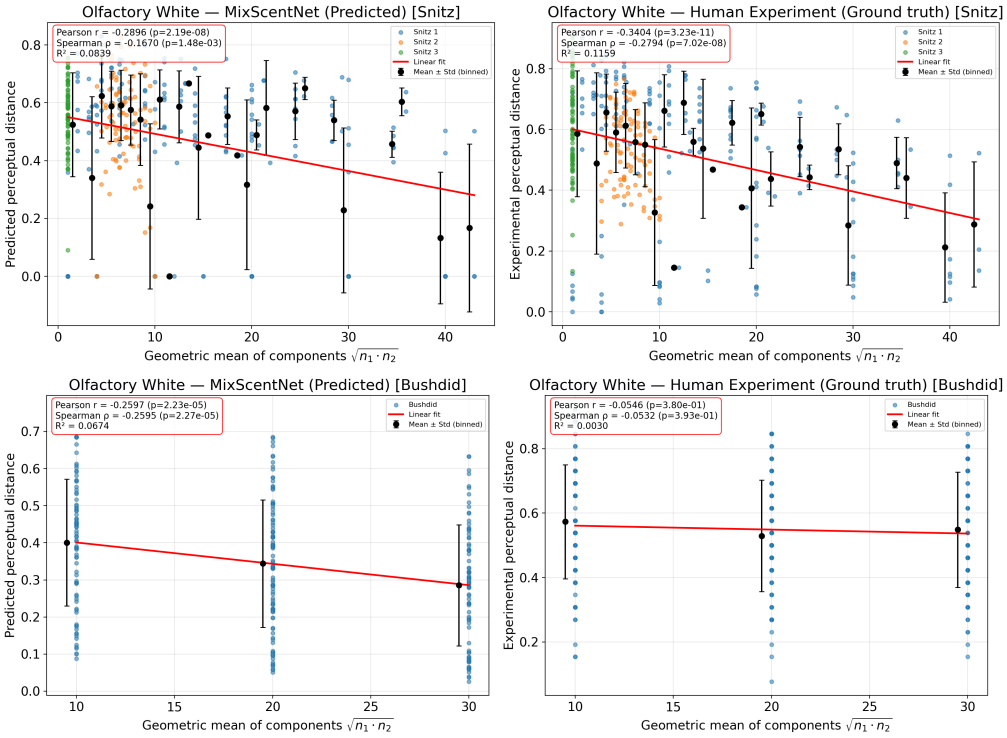

Figure A.3: More *olfactory white* reproduced results on the Snitz [16] and Bushdid [34] subsets in DREAM2024 [32], where MixScentNet is trained on [Ravia, Bushdid] and [Snitz, Ravia], respectively.

A.3 Dataset details

Pretraining the PubChem [18] dataset. The raw data are provided by Green *et al.* [8], comprising a total of 123,852,144 SMILES strings randomly collected from PubChem [18]. We further filter this large volume of SMILES strings using the following criteria:

- **SMILES string length.** We discard SMILES strings with more than 150 characters. This is a purely string-level operation that reduces the computational cost of RDKit [19] and the Mordred [20] descriptor calculator. Shorter SMILES strings also accelerate the D-MPNN [17] training process.
- **Multicomponent SMILES.** We discard SMILES strings containing the dot character (.), which is also a string-level operation and does not involve chemical analysis. The dots in a SMILES string indicate a multicomponent entity, including salts, ion pairs, solvates, and eutectics, which are not useful because we focus on single-molecule feature extraction.
- **Inorganic atoms.** We discard SMILES strings containing any of the following inorganic atoms: `[He, Na, Mg, Al, K, Ca, Ti, V, Cr, Fe, Co, Cu, Zn, Bi]`. Molecules containing these atoms are typically nonvolatile under ambient conditions or are toxic and therefore unsuitable for olfactory perception tasks.
- **RDKit "`MolFromSmiles(SMILES)`" validity verification.** We discard any SMILES string for which "`MolFromSmiles(SMILES)`" in the RDKit toolkit [19] returns "`None`". This filters out nonstandard SMILES syntax that RDKit cannot analyse, molecules containing unsupported atom types, and strings with syntax errors.
- **Valence contradictions and anomalous hydrogen counts.** We further execute "`RemoveHs(mol, updateExplicitCount=True)`" in the RDKit toolkit [19]; if any error is raised, the corresponding SMILES string is discarded. This process filters out molecules with contradictory valence bonds or abnormal hydrogen counts, *i.e.*, boundary cases that RDKit can represent but cannot process normally.

- **Molecular weight** We retain only molecules with weights from 30–300 Da, which is directly inspired by Mayhew *et al.* [38]. They reported that molecules from 30–300 Da typically have an odor. They demonstrate this in their proposed dataset, where 96% of molecules meeting these criteria were confirmed to have an odor.

With these criteria, we finally derive the refined PubChem dataset for pretraining, with a total of 1,168,920 SMILES.

DREAM 2025 [31] and DREAM 2024 [32]. The 51 perceptual labels of DREAM2025 are listed as follows.

```
Green, Cucumber, Herbal, Mint, Woody, Pine, Floral, Powdery, Fruity,
Citrus, Tropical, Berry, Peach, Sweet, Caramellic, Vanilla, BrownSpice,
Smoky, Burnt, Roasted, Grainy, Meaty, Nutty, Fatty, Coconut, Waxy,
Dairy, Buttery, Cheesy, Sour, Fermented, Sulfurous, and Garlic.  Onion,
Earthy, Mushroom, Musty, Ammonia, Fishy, Fecal, Rotten.  Decay, Rubber,
Phenolic, Animal, Medicinal, Cooling, Sharp, Chlorine, Alcoholic,
Plastic, Ozone, Metallic
```

Each mixture is rated on these descriptors using the rate-all-that-apply (RATA) method by trained panelists. During panelist training, each descriptor was paired with visual and odor references. Only subjects who met the performance standards on a pretest of 20 common odourants were invited to join the panel: the individual test-retest correlation was $R > 0.35$, and reasonable labels for these odourants were selected. The DREAM2025 [31] organizers did not disclose how the final perceptual scores were aggregated across panelists or how the scent mixtures were physically prepared from the constituent molecules.

DREAM2024 [32] aggregates mixture pairs from three prior studies, Snitz *et al.* [16], Ravia *et al.* [33], and Bushdid *et al.* [34], which differ in both experimental objectives and perceptual distance measurement. Bushdid *et al.* [34] aimed to quantify the upper bound of human olfactory discrimination, employing a triangle test paradigm: the three samples were presented on each trial, where two from the same mixture and one from a different mixture. The participants were required to identify the different ones. To ensure that the discrimination was based on odor quality rather than intensity, the three mixtures were presented at different dilutions (undiluted, 1:2, and 1:4), and the perceptual distance for each pair was defined as the proportion of successful discriminations. Ravia *et al.* [33] was designed to construct olfactory metamers, *i.e.,*, mixture pairs with disjoint molecular compositions but perceptually indistinguishable odors. Participants were required to rate the perceived similarity of two mixtures on a continuous scale. Snitz *et al.* [16] focused on establishing a chemical-descriptor-based odor similarity metric and collected perceptual distances through direct similarity ratings. Since these studies differ in terms of paradigm and sample presentation, the DREAM2024 organizers unified all the measurements onto a continuous 0–1 perceptual distance scale via linear normalization. Specifically, the triangle test success rates naturally fell within this range and were used directly, while direct similarity ratings were normalized accordingly. The DREAM2024 organizers [32] also acknowledge that their unification cannot fully correct for paradigm-specific biases or systematic noise from different delivery methods (*e.g.*, headspace delivery via pads-in-jar [33] versus liquid in vials [16, 34]) and that these residual inconsistencies represent an inherent limitation of the unified dataset.

A.4 D-MPNN implementation and pretraining details

D-MPNN implementations. An overview of the D-MPNN implementations and the pretraining process is depicted in Figure A.4. We start by elaborating on processing the input SMILE string. Each SMILES string is converted to a molecular graph via two specific functions "`MultiHotAtomFeaturizer.v2`" and "`MultiHotBondFeaturizer`" [17]. Here, the atom features (dimension $d_v = 72$) encode the atomic number, total degree, formal charge, chirality tag, hydrogen count, hybridization, aromaticity, and normalized atomic mass. Bond features (dimension $d_e = 14$) encode bond type, conjugation, ring membership, and stereochemistry. Each chemical bond is expanded into two directed edges, and a reverse-edge index is maintained to exclude back-flowing messages during aggregation. We adopt the D-MPNN [17] with a hidden dimension of $d_h = 2048$

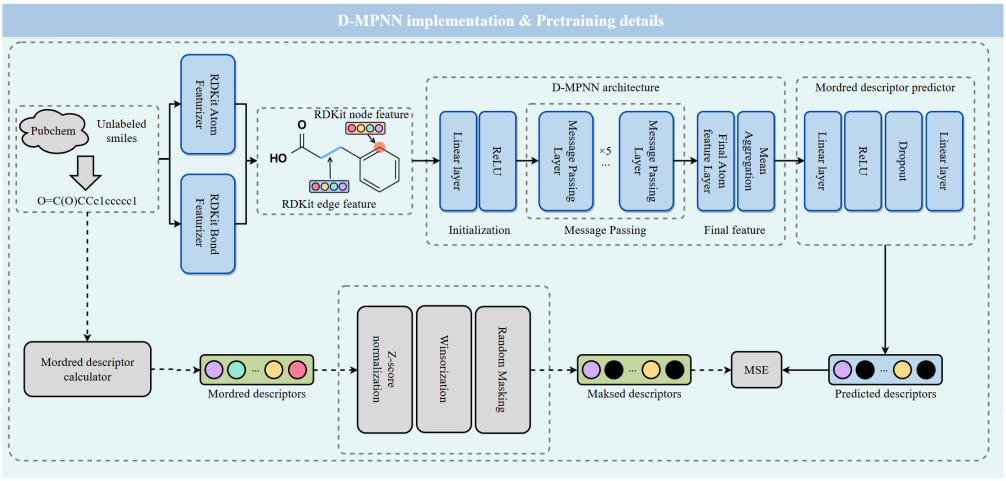

Figure A.4: Overview of the D-MPNN and the pretraining details.

and a depth of $T = 6$. For each directed edge $(v \to w)$, the graph messages are updated as follows:

$$h_{vw}^{(0)} = \mathrm{ReLU}(W_i \cdot [\mathbf{x}_v \,\|\, \mathbf{e}_{vw}])\,, \tag{12}$$

$$h_{vw}^{(t)} = \mathrm{ReLU}\left(h_{vw}^{(0)} + W_h \cdot \sum_{u \in \mathcal{N}(v)\setminus\{w\}} h_{uv}^{(t-1)}\right), \quad t = 1, \ldots, T-1, \tag{13}$$

$$h_v^{(T)} = \mathrm{ReLU}\left(W_o \cdot \left[\mathbf{x}_v \,\|\, \textstyle\sum_{w \in \mathcal{N}(v)} h_{vw}^{(T-1)}\right]\right), \tag{14}$$

where $W_i \in \mathbb{R}^{(d_v+d_e)\times d_h}$, $W_h \in \mathbb{R}^{d_h \times d_h}$, and $W_o \in \mathbb{R}^{(d_v+d_h)\times d_h}$ are weight matrices of the Linear layers in the initialization, the message passing and the final feature phase of the D-MPNN, respectively. Finally, the molecule-level feature is obtained by mean pooling over all the atom features, $\mathbf{z} = \frac{1}{|V|}\sum_{v \in V} h_v^{(T)}$. $\mathbf{z}$ is the only feature transferred to the downstream mixture model. During pretraining, a two-layer regression head maps $\mathbf{z}$ to the Mordred descriptor, which is composed of linear layers with ReLU activations and dropout, as shown in Figure A.4. This head is discarded after pretraining.

The pretraining details. We start by introducing the self-supervised target. We first compute the raw modified descriptors via the Mordred toolkit [20], which takes only the SMILES string as the input. The resulting descriptors are first Z-score normalized across each descriptor dimension rather than per sample, using the mean μ and standard deviation σ computed on the training split. Winsorization is applied by clamping to $[-6\sigma, +6\sigma]$ to suppress the influence of extreme outliers. We further adopt the masking strategy adopted in ChemBERTa [29]: for each training sample, every descriptor dimension is independently masked at a rate of 15%, and the MSE loss is computed between the masked descriptors and the predicted descriptors. The mask is recomputed every batch and every epoch, leading to effective data augmentation. The pretraining randomly splits the refined PubChem dataset [18] into 70%/20%/10% for training, validation, and testing, where the batch size is 128. We adopt the Adam optimizer and train for up to 500 epochs with early stopping on validation loss (patience $= 50$). The top-2 checkpoints ranked by validation loss are retained, and the best one is used to initialize the downstream encoder.

A.5 Implementation details of baseline methods

For the classic machine learning-based models, we directly utilized the realization proposed by Gray *et al.* [12], including the Snitz baseline [16] and XGBOOST with diverse feature embeddings. Details can be found in their open-source repository `https://github.com/chemcognition-lab/pom-mix`. Some of the compared deep learning methods are not proposed for mixed olfactory prediction tasks, and we modify them on the basis of their original open-source implementations.

Table A.2: Comparison between the original MolSets [7] and our modified version used as a baseline. **Left:** retained designs based on the original architecture. **Right:** modified parts.

Designs	Retained
Graph construction	✓
GNN backbone	✓
Global meanpool	✓
Activations: tanh	✓
MLP prediction head	✓

(a) Retained designs.

Designs	Original MolSets	Modified (ours)
Physical inputs	MW, frac, salt, molality	Graphs only
Encoders	2 (Solution+salt)	1 (Mixture)
Attention	Single-head	Multihead
Output modes	Regression only	Regression + distance
Applicability	Narrow (fixed prediction)	Broad (multitask)

(b) Modified designs.

MolSets [7]. This model was originally designed to predict the ionic conductivity of polymer electrolyte systems and has two main parts: a network for processing solution mixtures and a network for processing salts. The solution component consists of a single-molecule encoder and a permutation-invariant predictor. The encoder stacks three SAGEConv [39] layers, each followed by LayerNorm and ReLU, a global mean pooling, and a linear layer that concatenates the molecular weights (MW) into the graph embedding. A single-head self-attention module is used to capture molecular interactions in solution, and the attention outputs are aggregated into a mixed representation via mass-fraction weighting. For the salt component, the structurally identical but weight-independent salt encoder produces the salt embedding, which is concatenated with the solution embedding and input into the prediction head. To adapt MolSets to our olfactory tasks, we apply several modifications while preserving the core designs of the original backbone. The detailed modifications and preservation are presented in Table A.2b. Specifically, the main modifications focus on the encoders and the output heads, and minor modifications introduce multihead self-attention. For the encoders, we adopt the solution GNN architectures of MolSets to process mixtures and remove the salt encoder. We also modified the input of the GNN to take only the component molecular graphs as input. For the output head, we modify the label prediction head to be the same as our MixScentNet and use the cosine distance to compare the mixture-level features for the perceptual distance task.

Chemeleon [8] also adopts self-supervised pretraining with MixScentNet but is designed as a single-molecule encoder and cannot directly produce mixture-level representations. We include it as a baseline to assess whether the naïve aggregation of pretrained single-molecule features can rival methods specifically designed for mixture tasks. We keep the CheMeleon backbone unchanged and attach two task-specific output heads: the same MLP head as MixScentNet for label prediction and a "feature-summation + cosine-distance" for perceptual distance estimation. The backbone is fine-tuned end-to-end on both downstream olfactory tasks. For the POMMix [12], we evaluate it without any modification and apply it on the same training/testing splits as MixScentNet does.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have included them in the abstract.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question is not perceived well by the reviewers.
   - The claims made should match theoretical and experimental results and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation, while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when the image resolution is low or when images are taken under low lighting. Alternatively, a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that are not acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favour of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: No assumptions or proofs in our work.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided at the core of the paper should be complemented by formal proofs provided in the appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose the main code in the anonymous link: `https://anonymous.4open.science/r/Odor-D2DB`

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example,
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should be either a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, access to the model may be limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproduce or verify the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material?

Answer: [Yes]

Justification: Access to all the databases is shown in Appendix A.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in the supplemental material (appended to the paper) is recommended, but including URLs in the data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have specified all the details in Appendixes A.4 and A.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in the appendix, or as supplemental material.

7. **Experimental statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the mean and standard deviation of the 5-fold cross-validation results and the p-value in reproducing the olfactory white results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the details of the computing resources in Section 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute worker CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the number of computations required for each of the individual experimental runs as well as estimate the total number of computations.
   - The paper should disclose whether the full research project required more computations than the experiments reported in the paper did (e.g., preliminary or failed experiments that did not make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that we respect the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both the potential positive societal impacts and the negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The social impact of our work is positive, and we discuss the positive social impact in Section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not necessary to note that a generic algorithm for optimizing neural networks could enable people to train models that generate deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not have such a risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual use should be released with necessary safeguards to allow for controlled use of the model, for example, by requiring that users adhere to usage guidelines or restrictions to access the model or implement safety filters.
    - Datasets that have been scraped from the internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licences for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the licence and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all the data are under proper licence.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the licence (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the licence, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licences for some datasets. Their licensing guide can help determine the licence of a dataset.
    - For existing datasets that are repackaged, both the original licence and the licence of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not have new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, licences, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose assets are used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing experiments or research is considered in our work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing or research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No such issue.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing or research with human subjects.
    - Depending on the country in which the research is conducted, IRB approval (or equivalent) may be required for any human subject research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this process may vary significantly across institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or nonstandard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, a declaration is not needed.

Answer: [N/A]

Justification: We do not use LLM in our work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or nonstandard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.