An electronic nose (e-nose) mimics the mammalian olfactory system in identifying odors and expands human olfaction boundaries by tracing toxins and explosives. However, existing feature-based odor recognition algorithms rely on domain-specific expertise, which may limit the performance due to information loss during the feature extraction process. Inspired by human olfaction, a smart electronic nose enabled by an all-feature olfactory algorithm (AFOA) is proposed, whereby all features in a gas sensing cycle of semiconductor gas sensors, including the response, equilibrium, and recovery processes are utilized. Specifically, our method combines 1D convolutional and recurrent neural networks with channel and temporal attention modules to fully utilize complementary global and dynamic information. It is further demonstrated that a novel data augmentation method can transform the raw data into a suitable representation for feature extraction. Results show that the e-nose simply comprising of six semiconductor gas sensors achieves superior performances to state-of-the-art methods on the Chinese liquor data. Ablation studies reveal the contribution of each sensor in odor recognition. Therefore, a deep-learning-enabled codesign of sensor arrays and recognition algorithms can reduce the heavy demand for a huge amount of highly specialized gas sensors and provide interpretable insights into odor recognition dynamics in an iterative way.

1. Introduction

Humans can create their perception of the world through sight, hearing, touch, olfaction, and taste. Olfaction is important for the survival of living species and allows living species to be able to identify suitable food, detect dangerous chemical substances, etc. The olfactory system of humans is based on a chemical reaction that is more complicated than the physical stimulus in the vision and auditory systems. Odors are complex and always contain various types of gas molecules. For mammalian olfaction, each olfactory receptor cell possesses only one type of odorant receptor, and each receptor can detect a limited number of odorant substances. Our olfactory receptor cells (ORCs) are, therefore, highly specialized for a few odors. Namely, each odor molecule activates very few odorant receptors, leading to a combinatorial code and forming an “odor pattern.” Based on a large number of receptors and complex neural networks, we can discriminate more than one trillion olfactory stimuli.

Inspired by the nature of the mammalian olfactory system, a “mode nose” was first introduced for gas identification in 1982. It contains a semiconductor gas sensor array that mimics the function of mammalian ORCs with a pattern recognition algorithm to simulate the operations of the nervous system. The non-specific semiconductor gas sensor detects a certain gas from a change in electrical resistance caused by the reaction between gas molecules and preabsorbed oxygen, thereby having cross-sensitivity to a wide variety of odors. In 1987, Kaneyasu et al. from Hitachi in Japan named it an “electronic nose” (e-nose), and e-noses were introduced into many fields in the 1990s. The e-nose has shown great potential for expanding the human sense, especially in recognizing gases with no flavor or low concentrations, which may find wide applications in environmental monitoring, food quality assessment, medical diagnosis, etc.

Unfortunately, to date, an e-nose mimics the mammalian olfactory system at a gross level. This is because semiconductor gas sensors are far behind ORCs in specificity, diversity, and scale. Therefore, a powerful pattern recognition algorithm to handle complex gas–solid interactions under limited hardware conditions is needed.

Generally, traditional feature-based methods are all multi-stage, including feature extraction, dimensionality reduction, and classification. In feature extraction, some bioinspired
or manually designed features \cite{17, 18, 19} are extracted from the response curves based on a basic understanding of the gas sensing mechanism, which mainly contains equilibrium statuses such as resistance values, response/recovery times, and the maximum derivation of the response times. In dimensionality reduction, variants of principal components analysis (PCA) are often used. \cite{20, 21} Finally, existing feature-based methods use unsupervised learning \cite{22, 23} and backpropagation artificial neural networks (BP-ANNs) for classification. \cite{21, 24, 25, 26} These feature-based methods mainly contain equilibrium statuses while neglecting response and recovery features, which may lead to local optima and information loss, before feeding them to the classifier. Therefore, the features extracted from the whole gas sensing curves, including the response, equilibrium, and recovery processes, can play an important role in odor recognition. Humans have only one-third as many types of olfactory receptors as mice but have superior processing power due to stronger brain connections, \cite{27} and some studies have even found that odors can affect cognition. \cite{28, 29} Enabled by the power of deep learning, the focus of this study is on the accurate recognition of various odorant mixtures using only a small number of sensor units combined with an all-feature extraction algorithm in a complex circumstance (uncontrolled temperature and humidity). We hypothesize that all features in a gas sensing cycle of the sensor array can produce more distinguishing and robust features, thus reducing the heavy demand for the quantity and diversity of sensors. Hence, we need a more effective algorithm for application-specific sensing scenarios.

Recently, deep-learning-based methods have exhibited surprising progress in computer vision, natural language processing, medical imaging, etc. \cite{30} Unlike feature-based methods that heavily rely on intuition or domain-specific experience, deep-learning-based methods attempt to learn high-level semantic features from mass data and jointly optimize feature extractors and classifiers to significantly decrease the burden on users. Introducing deep-learning-based methods to e-nose technology can improve the performance by learning nonintuitive features with deep learning. In addition, the learned features can also help us understand the principle of gas sensing and odor discrimination. Recently, some researchers have treated multichannel response curves as an image and used 2D convolutional neural networks (CNNs) to extract local features and fully connected (FC) layers for classification in an end-to-end manner. \cite{31, 32} Although these methods achieve performance improvements over feature-based methods, they ignore the long-term dependencies in time-series signals of the sensor array and bring non-negligible computational and memory overhead. Wang et al. \cite{33} proposed a quantitative detection method of mixed gases based on long short-term memory (LSTM). This method heavily relies on domain-specific expertise, as the preprocessed response data to be analyzed are manually designed, which may lead to information loss before they are fed into LSTM. However, deep learning applied to raw data can help to better mine cross selectivity among only a few sensors. To tackle the aforementioned issues, we fabricate an e-nose that consists of six different metal–oxide–semiconductor (MOS) gas sensors, including SnO$_2$ quantum dots (QDs), SnO$_2$ nanowires, SnO$_2$ nanoparticles (NPs) synthesized by flame spray pyrolysis (FSP), In$_2$O$_3$ QDs, NiO NPs, and WO$_3$ QDs. MOS favors the e-nose due to its high response rate, low cost, easy fabrication, and long-term stability. In particular, QDs are critical low-dimensional semiconductor materials, \cite{34, 35} whose dimensions in three axes are not larger than twice the exciton Bohr radius. To reduce the heavy demand for the number and diversity of sensors, we use tailored data augmentation to handle all features in a gas sensing cycle and therefore transform the raw curves into different shapes to simplify distinguishing and robust feature mining. Specifically, our method combines 1D CNNs and recurrent neural networks (RNNs) with channel and temporal attention modules to fully utilize complementary global and dynamic information in an end-to-end manner. We also demonstrate the generalization power of this data augmentation process, which can significantly improve the performance of feature-based methods. It is worth noting that the people who performed the measurements were not well trained, i.e., experimental errors were introduced into the data, which is similar to practical application scenarios. By consisting of only six nonspecific semiconductor gas sensors, the all-feature olfactory algorithm (AFOA)-enabled e-nose can discriminate these Chinese liquors with high accuracy. It can also be concluded that QDs are superior to other sensors.

2. Experimental Section

2.1. E-Nose Design and Measurement

As shown in Figure 1, the e-nose utilizes a semiconductor gas sensor array to mimic the function of mammalian ORCs. Combined with the pattern recognition algorithm, an e-nose can discriminate different odors. Based on this principle, an e-nose with a gas sensor array of 6 nanostructure MOS sensors, a data acquisition card, and gas sampling was fabricated (Figure 2a, b). The raw response curves of our e-nose for five similar Chinese liquors and air are depicted in Figure 2c. All the sensors reached saturation when the response to the vapor is about 20 s. It is worth noting that the people who performed the measurements were not well trained, i.e., experimental errors were introduced to the data, which is similar to practical application scenarios.

2.2. E-Nose Odor Recognition Algorithm Design

Unlike other deep-learning-based methods that directly use a 2D convolution kernel, we consider the response curves of the sensor array as a multichannel 1D time-series signal and thus use a 1D ResNet-like \cite{46} network as our backbone to extract all features. The overall architecture of our all-feature olfactory algorithm (AFOA) is shown in Figure 3a. The ResNet-like network first involves a 1D convolutional layer and three duplicate stages, and each stage is stacked with multiple residual bottlenecks. Figure 3b shows that a standard residual bottleneck is chosen to stack $1 \times 1$ and $3 \times 1$ convolutional layers with batch normalization (BN) \cite{57} and ReLU. \cite{38} This bottleneck architecture contains two $1 \times 1$ convolutional layers for dimension reduction and raising. It is computationally efficient and can implement information interaction and fusion across different channels.
The convolutional block attention module (CBAM)\(^{[39]}\) is a technique used in the feature map of 2D convolution, and we adapt it to the 1D situation. The module has two submodules: channel and temporal modules. Figure 3c shows that the intermediate feature map is adaptively refined through CBAM at every residual block of the feature extraction network before being added to the identity shortcut connection.

After the backbone, we implement a two-branch network. It is composed of a global feature block (GFB) and a dynamic modeling block (DMB), which can learn global features and temporal information simultaneously. In GFB, we compute the average value of all values across the entire matrix for each of the input channels by global average pooling, which is inspired by the feature extraction step of handcrafted methods. Therefore, we obtain a feature vector of length 512, each value of which represents a specific feature extraction method. Simultaneously, the output of the backbone network is conveyed into an average pooling layer to further expand the receptive field, thus extracting high-level semantic information. The semantic time-series signal is then passed into the LSTM block. The LSTM block can model long-term dependencies, which are hard to learn with CNNs.

Finally, the outputs of the two branches are concatenated and then passed through two stacked FC layers. A softmax classification layer is added at the top of the network to output the probability of every category.

### 2.3. Sensor Array

The sensor array was composed of SnO\(_2\) QDs, SnO\(_2\) nanowires, SnO\(_2\) FSP, In\(_2\)O\(_3\) QDs, NiO NPs, and WO\(_3\) QDs. Details of the synthesis of the sensing materials are described in the Supporting Information.

The ceramic plate substrate (1.0mm × 1.5mm) was previously coated with gold electrodes and ruthenium oxides as the heater. Gas sensing materials were coated on ceramic substrates by dripping. The four electrodes of the ceramic chip were then welded to the base to form a single gas sensor element. Annealing was used to enhance the stability of each sensor. The operating temperature of the sensors was set to 300 °C.

### 2.4. Acquisition System

The e-nose consisted of a gas inlet, a power source, a sensor array chamber, a micropump, and a data acquisition unit. The structure of the e-nose and the morphology of the sensing materials are shown in Figure 2a,b and S3, Supporting Information.

Five different Chinese liquors, ChunGuJiu (CGJ), BaiYunBianJianXiang (BYBJX), BaiYunBianNongXiang (BYBNX), SiTeJiu (STJ), and MaoTaiWangZi (MTWZ), were chosen for the benchmark. The contents of the Chinese liquors are very complex, primarily including ethanol and various kinds of esters. Each brand has a unique fragrance by the making process, infusing alcohol with other ingredients, such as herbs and spices. The compositions and photos of the five liquors are listed in Table S1 and Figure S2, Supporting Information, respectively.

10 mL of Chinese liquor was put into a glass bottle, and waited for 5 mins to reach evaporating equilibrium at room temperature. In the beginning, the e-nose was stabilized for 40 s in the ambient air, and then the sample was placed at the inlet of the e-nose. The headspace of the sample was vacuumed into the chamber for the test (see Figure S4, Supporting Information). After sensing for about 20 s, the sample was removed, and...
recovered for 120 s. During the test, the flow rate was maintained at 100 mL min⁻¹ by the pump in the E-nose. Each sample was tested twice a day, 50 times one month. The time interval between each test was half an hour to ensure that the sensors recover to the original baseline.

In the test, the ambient environment was not well controlled, where the temperature and relative humidity changes were 15±25 °C and 30±70% RH, respectively. Three operators who took part in the measurement were not well trained, i.e., experimental errors were introduced to the data, which is similar to practical application scenarios. CGJ: ChunGuJiu, BYBJX: BaiYunBianJianXiang, BYBNX: BaiYunBianNongXiang, STJ: SiTeJiu, MTWZ: MaoTaiWangZi.

Figure 2. Schematic illustrations of the e-nose. a) The structure of the e-nose. b) The morphology of the sensing materials. The number in the lower right corner of the image denotes the sensor channel. QDs: quantum dots, FSP: nanoparticles synthesized by flame spray pyrolysis, NP: nanoparticles. c) The raw response curves of the metal–oxide–semiconductor (MOS) sensor array for five Chinese liquors and air. It is worth noting that the people who performed the measurements were not well trained, i.e., experimental errors were introduced to the data, which is similar to practical application scenarios. CGJ: ChunGuJiu, BYBJX: BaiYunBianJianXiang, BYBNX: BaiYunBianNongXiang, STJ: SiTeJiu, MTWZ: MaoTaiWangZi.
2.5. Attention Mechanism

Given an intermediate feature map \( F \in \mathbb{R}^{C \times L} \) as input, CBAM sequentially infers a 1D channel attention map \( M_c \in \mathbb{R}^{C \times 1} \) and a 1D temporal attention map \( M_T \in \mathbb{R}^{1 \times L} \), as illustrated in Figure 3c. The overall attention process can be summarized as

\[
F = M_c(F) \otimes F \\
F'' = M_T(F') \otimes F'
\]

where \( \otimes \) denotes elementwise multiplication. During multiplication, the attention values are broadcast accordingly: channel attention values are broadcast along the time dimension, and vice versa. \( F'' \) is the final refined output. Figure 3d,e depicts the computation process of each attention map in detail. Since CBAM does not change the feature map’s shape, we can arrange the two submodules as sequential channel-temporal, as sequential temporal-channel, or in parallel. It can be easily plugged into the current mainstream CNN.

2.5.1. Channel Attention Module

As shown in Figure 3d, we first aggregate the temporal information of the input feature map using both average pooling and max pooling to generate two different temporal context descriptors: \( F_{\text{avg}} \) and \( F_{\text{max}} \). Second, both descriptors are forwarded to a shared network to produce a channel attention map \( M_c \in \mathbb{R}^{C \times 1} \). The shared network is composed of a multilayer perceptron (MLP)
with one hidden layer, whose size is set to $R^{C \times 1}$, where $r$ is the reduction ratio. Finally, we merge the output feature maps of the shared MLP using elementwise summation. In short, the channel reduction ratio. Finally, we merge the output feature maps of the

\[
M_C(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) = \sigma(W_1(W_0(F_{\text{avg}}) + W_1(W_0(F_{\text{max}})))
\]

where $\sigma$ denotes the sigmoid function, $W_0 \in R^{C \times C}$ and $W_1 \in R^{C \times C}$ denote the MLP weights, and the ReLU activation function is followed by $W_0$.

2.5.2. Temporal Attention Module

Complementary to channel attention, temporal attention focuses on informative positions along the time axis. As shown in Figure 3e, we first apply average pooling and max pooling along the channel axis, generating two 1D maps, $F_{\text{avg}}^T \in R^{1 \times L}$ and $F_{\text{max}}^T = R^{1 \times L}$, and concatenate them to generate an effective feature descriptor. Then, we apply a 1D convolution layer to generate a temporal feature map $M_T(F) \in R^{L \times 1}$, which encodes where to emphasize or suppress. In short, temporal attention is computed as

\[
M_T(F) = \sigma(f^{7 \times 1}(\text{AvgPool}(F); \text{MaxPool}(F))) = \sigma(f^{7 \times 1}(F_{\text{avg}}^T; F_{\text{max}}^T))
\]

where $\sigma$ denotes the sigmoid function and $f^{7 \times 1}$ represents a 1D convolution operation with a filter size of $7 \times 1$.

2.6. Global Feature Block

The feature maps extracted by 1D ResNet have lengths of 195 pixels with 512 channels. Inspired by the handcrafted feature-based method, we obtain the average value in the 1D feature map through global pooling. Therefore, we obtain a feature vector of length 512 and each value of which represents a specific feature extraction method.

2.7. Dynamic Modeling Block

Simultaneously, the output of 1D ResNet is conveyed into an average pooling layer to further expand the receptive field, thus extracting high-level semantic information. The semantic time-series signal is then passed into a standard LSTM network with dropout for a linear transformation of the recurrent state. Unlike a CNN, which can only model local features, LSTM has a chain-like structure capable of learning long-term dependencies. Instead of having a single neural network layer, the repeating module in LSTM contains four interacting layers. LSTM maintains a hidden vector $h$ and a memory vector $C$, which are responsible for controlling state updates and outputs at each time step. More concretely, Graves et al. defined the computation at time step $t$ as follows

\[
\begin{align*}
\{f_t, i_t, \tilde{C}_t, o_t, h_t\} &= \{\sigma(W_f[h_{t-1}, x_t] + b_f), \\
i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \quad &\tilde{C}_t &= \tanh(W_C[h_{t-1}, x_t] + b_C), \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad &o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\
h_t &= o_t \odot \tanh(C_t)
\end{align*}
\]

where $\sigma$ is the logistic sigmoid function; $\odot$ represents element-wise multiplication; $W_f$, $W_i$, $W_C$, and $W_o$ are the recurrent weight matrices; and $b_f$, $b_i$, $b_C$, and $b_o$ are the recurrent offset vectors. The outputs of GFB and DMB are concatenated and then passed through two stacked FC layers. The dimension of each FC layer is 512. Finally, a softmax classification layer is added at the top of the network to output every category’s probability.

2.8. Implementation Details

We implement and evaluate different deep-learning-based architectures, as summarized in Table 1. We adopt BN and ReLU activation after each convolution sequentially, following 2D ResNet. Dropouts of 0.3 are applied before and after LSTM with 64 hidden layers. Dropouts of 0.5 are applied in the two FC layers. The network is trained by Adam with a batch size of 64 on a single NVIDIA GeForce RTX 2080 Ti GPU. The initial training rate is $10^{-3}$, which is divided by ten at iterations 80, 120, and 180, and training stops after 200 iterations. We do not use dropout in the test phase. The proposed network is implemented using Python (Version 3.6 with Tensorflow 1.13 and Keras 2.2). For feature-based methods with hand-crafted features, the number of PCA components $k$ is set from 20 to 60 to derive the best feature space dimension. For feature-based methods with data augmentation, the number of PCA components $k$ is set from 20 to 150. We apply grid research for parameters $C$ and gamma of SVM, where

\[
C \in [1, 5, 10, 50, 1e2, 5e2, 1e3, 5e3, 1e4, 5e4, 1e5, 5e5, 1e6] \quad \text{(6)}
\]

\[
\text{gamma} \in [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5] \quad \text{(7)}
\]

and SVM kernel $\in [\text{linear, poly, rbf, sigmoid}]$. For deep-learning-based and feature-based methods, we perform fivefold cross-validation.

2.9. Implementation of Gramian Angular Summation Field

As the feature maps to be visualized are all 1D vectors defined as $X = \{x_1, x_2, \ldots, x_n\}$, we first rescale $X$ so that all values fall in the interval $[-1, 1]$ by

\[
\hat{x}_i = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)}
\]

Thus, we can represent the rescaled vector $X$ in polar coordinates, in which the angle and radius in polar coordinates are defined as

\[
\begin{align*}
\varphi &= \arccos(\hat{x}_i), \quad -1 \leq \hat{x}_i \leq 1, \hat{x}_i \in \hat{X}, \\
r &= \frac{1}{K}, i \in N
\end{align*}
\]
where $t_i$ is the time stamp and $N$ is a constant factor used to regularize the span of the polar coordinate system.

Then, we can easily exploit the polar coordinate perspective to encode the trigonometric summation between each point to identify the correlation within different time intervals. The Gramian angular summation field (GASF) is defined as

$$\text{GASF} = \left[ \cos(\Theta_1 + \Theta_2) \right] = \tilde{X}^T \tilde{X} - \sqrt{1 - \tilde{X}^T \tilde{X}} \cdot \sqrt{1 - \tilde{X}^T \tilde{X}}$$  \hspace{1cm} (10)$$

where $I$ is the unit row vector $[1, 1, \cdots, 1]$. By defining the inner product as $\langle x, y \rangle = x^T y - \sqrt{1 - x^T x} \cdot \sqrt{1 - y^T y}$, GASF is actually a quasi-Gramian matrix $[[x_1 x_1], \cdots, [x_N x_N]]$. 

### 2.10 Statistical Analysis

#### 2.10.1 Preprocessing of Data

The MOS sensor array produces six-channel time-series signals, such as the one shown in Figure 2c. It is almost impossible for ordinary humans or even sensor experts to observe the distinguishability between different Chinese liquors. The intensity of the y-axis is the resistance of the MOS sensor from different sensors at a specific time point (x-axis). There are five raw n-type sensors and one raw p-type sensor in the sensor array. For the $n$-type sensor case, the falling edge indicates when the liquor to be measured enters the measuring chamber. In contrast, the slow recovery of the curve from the zero response to the baseline suggests that the liquor leaves the measuring chamber, and vice versa for the $p$-type sensor.

Due to the need for massive data in deep-learning-based algorithms, we augment the raw data according to the process shown in Figure S1, Supporting Information.

1) Data format. We denote our original dataset as $D = \{(T_r, y_r)\}_{r=1}^N$, which contains $N$ raw data samples and the corresponding labels. We denote each sample as $T_r = \{T_r^{(0)}, \cdots, T_r^{(K)}\}_{j=1}^K$, which contains $K$ time-series signals from their associated sensor units. For each original sample $T_r$, a slice is a snippet of the original time series and is defined as

$$S_{t:i,j} = \{T_r^{(0)}, \cdots, T_r^{(K)}\}_{j=1}^{K}, 1 \leq i \leq j \leq n$$  \hspace{1cm} (11)$$

where $i$ denotes the time label at the beginning of the window slice and $j$ denotes the time label at the end of the window slice.

Given a time series $T_r$, of length $n$ and the slice length $s$, our slicing operation generates a set of $n - s + 1$ sliced time series

$$\text{Slicing}(T_r, s) = \{S_{1:s}, S_{2:s+1}, \cdots, S_{n-s+1:s}\}$$  \hspace{1cm} (12)$$

where all the time series in the Slicing($T_r, s$) have the same label as their original time series $T$. Thus, the sample number of the original dataset is expanded to $N \times (n - s + 1)$.

2) Reciprocal transformation and dataset partition. After slicing, we experimentally perform a reciprocal transformation on sensor four and sensor six

$$S_{t=i}^{k} = \frac{1}{S_{t=i}}$$  \hspace{1cm} (13)$$

where $k$ denotes the sensor number requiring a reciprocal transformation.

3) Normalization. To eliminate the influence of material response scale differences and baseline drift on classification performance, normalization of $S'_t$ is performed by

$$S''_t = \frac{S'_t - \min(S'_t)}{\max(S'_t) - \min(S'_t)}$$  \hspace{1cm} (14)$$

where $S'_t$ denotes each sensor’s response at time $t$. It is clear that after the implementation of normalization, the response curves are scaled to the range $[0,1]$. Thus, curves from different sensors are comparable, which is beneficial for subsequent feature extraction.

Then, for training and testing, a random shuffle 80–20% split is used in each category. Furthermore, we apply data augmentation to increase the robustness of our model and compensate for the baseline drift of the sensor array. For the training set, the resistant values in each sensor are increased by random values in the range of 0.01% to 9.99%. Such a data augmentation operation changes the absolute value of the response intensity.
without changing the curve shape. Thus, the training set is augmented five times.

4) Multichannel input mapping. Since the principal component, ethanol, in Chinese liquors is present in dominant quantities, the response curves from different Chinese liquors typically have similar patterns in their shapes. Thus, most fine-grained features may be too small to be captured. To overcome this inherent problem, we perform a power function transformation on each value $x$ in the channels to map it to five distinct values. To the best of our knowledge, power function transformation was first used for data augmentation in gas chromatography-mass spectrometry (GC-MS).[^45] The function is defined by

$$ y(f(x)) = [x, x^{a_1}, \ldots, x^{a_p}], 0 < a_1 < \ldots < a_p < 1 $$  \hspace{1cm} (15)$$

where $p$ denotes the number of power functions. Thus, the number of channels in each sample is augmented from $K$ to $K \times (p + 1)$, where $K$ denotes the number of sensor units.

A visual inspection of a single sensor in Figure S5, Supporting Information, suggests that $p = 4$, $a_1 = 0.2$, $a_2 = 0.4$, $a_3 = 0.6$, and $a_4 = 0.8$ result in slow changes in the time dimension, making it easier for the network to extract local features. Finally, the number of samples in the training set is augmented 100 times. The number of samples in the testing set is augmented 20 times. The number of channels in both datasets is augmented five times.

2.10.2. Data Presentation

We randomly select 28 samples from each Chinese liquor and the baseline (air) to obtain a training set of 168 samples. We use the remaining 42 samples as the test set, with each category containing seven samples.

For feature-based methods without data augmentation, the training set consists of 168 segments, and the test set consists of 42 segments. After feature dimension reduction implemented by PCA, each segment with 6 channels has a shape of $10 \times 6$ or $20 \times 6$. The former 10 hand-crafted features are calculated during the whole sampling phase, while the latter 20 hand-crafted features are separately calculated during the adsorption and desorption phases.

For feature-based methods with data augmentation and deep-learning-based methods, the training set consists of 16800 segments, and the test set consists of 840 segments after the implementation of the procedure described in 5.8.1. Each segment has a shape of $779 \times 30$. See Figure S7, Supporting Information, for details.

2.10.3. Software used for Statistical Analysis

The significance level of the performance difference between our method and others is obtained based on statistical tests. The Wilcoxon signed-rank test is utilized. $P$-values $<0.05$ are considered to be statistically significant. The statistical analysis is performed using the Python libraries NumPy,[^46] scikit-learn,[^47] and SciPy.[^48] We demonstrate the recognition performance improvement in the presented approach by gradually adding different modules and their combinations to the baseline, which shows the effectiveness of each module. Table 1 summarizes the experimental results on two particular feature blocks and different attention arrangement methods.

From a multichannel time-series signal viewpoint, GFB is applied in the channels, while DMB works temporally. In addition, it is natural to think that we may combine two blocks to build a hybrid architecture. In this article, both feature blocks are applied in parallel, and then the outputs of the two blocks are concatenated and followed by two FC layers. We use the 1D ResNet-like network as the baseline CNN and compare the performance of GFB, DMB, and their combination. In contrast with the baseline CNN, GFB yields a 4.8% (89.4% vs 84.6%) improvement in precision, DMB yields a 5.7% (90.3% vs 84.6%) improvement in precision, and the parallel structure yields a 7.7% (92.3% vs 84.6%) improvement in precision, as shown in Table 1.

In contrast, we compare three different arrangement methods for the channel and temporal attention submodules. In contrast with the channel-first attention sequential structure, spatial-first attention yielded a 1.8% (94.1% vs 92.3%) improvement in precision and parallel attention yielded a 1.4% (93.7% vs 92.3%) improvement in precision, as shown in Table 1. The results indicate that utilizing both attention modules is crucial, while the best-arranging strategy further improves performance.

3.2. Comparison of Feature-Based Methods

To compare the proposed deep-learning-based method, we also build a feature-based odor recognition approach that consists of three steps: feature selection, feature reduction, and classification. Ten conventional features[^21] are chosen according to the normalization of response curves $S_r$. The time $t_1$ denotes when $S_r$ reaches its maximum value, the root mean square is denoted by $RMS_S$, the arithmetic mean is denoted by $AM_S$, the geometric mean is denoted by $GM_S$, the harmonic mean of $S_r$ is denoted by $HM_S$, the maximum value of the first-order derivative of $S_r$ is denoted by $M'_S$, the integration of $S_r$ at time $t_2$ is denoted by $I_{S_r}$, and the mean curvature of $S_r$ is denoted by $GM_{curv}$.

These features are numbered $f_1$ to $f_{10}$ according to the above order for each sensor. Thus, we obtain 60 features for each odor sample, where the ten features of the same MOS cell are arranged together. For example, the features of sensor one are numbered 1–10, the features of sensor two are numbered 11–20, and so on.

As the number of features is still too large for the classifier, we use PCA for feature reduction and adopt linear discriminant analysis (LDA) or a support vector machine (SVM) for classification. To investigate the best performance of SVMs, we tested four different kernels: linear $K(u, v) = u \cdot v$, polynomial
$K(u, v) = (\gamma u \cdot v + \delta)^s$, radial basis function (RBF) $K(u, v) = \exp\left(-\frac{u \cdot v}{\sigma^2}\right)$, and sigmoid $K(u, v) = \tanh(\gamma (u, v) + \theta)$ kernels. Here, the linear kernel was best in most cases.

To the best of our knowledge, raw sensor time-series signals have never been fed directly into PCA. In this article, we convey augmented data as preprocessed in the Online Methods section into PCA to investigate whether data augmentation is also useful in the feature-based method. Since the input of PCA must be a 1D vector, we flattened augmented $779 \times 30$ multidimensional time series into a vector of length $23,370$ for dimensionality reduction, where 779 is the length of the time series and 30 is the number of dimensions.

The results of the feature-based methods and the deep-learning-based method of different input data lengths are shown in Table 2. Handcrafted feature extraction or data augmentation is used for data preprocessing. In addition, SVM or LDA is used for classification. The results indicate that data augmentation is also beneficial for feature-based methods. For instance, when data augmentation is used, the relative improvements in average precision are 32.2% (79.7% vs 60.3%) and 16.6% (80.7% vs 69.2%), respectively. Furthermore, the proposed deep-learning-based method, which combines GFB and DMB, with full-length input data achieves an average precision of 94.1% and outperforms the best feature-based method (which has an average precision of 80.7%) by a large margin. The quantitative results in Table 2 also show that the prediction result improves as the length of the input data increases, which further supports our all-feature extraction algorithm.

### 3.3. Efficiency of Our Method

We compare the efficiency of our method with the 2D convolution kernel version. First, we replace all the $3 \times 1$ convolution kernels with $3 \times 3$ kernels in the first convolutional layer and each residual block of the backbone. Then, we replace the temporal attention module (a $7 \times 1$ convolution kernel) with a spatial attention module (a $7 \times 7$ convolution kernel). Furthermore, the size of the input signal is also $779 \times 30$. During the training phase, the 2D convolution kernel version consumes $8.9$ G of GPU memory, and the training time is $\approx 6.3$ h with CBAM and $3.3$ h without CBAM, whose average precision results are 79.6% and 86.1%, respectively. In contrast, the 1D convolution version consumes $4.8$ GB of GPU memory, and the training time is $2.2$ h with CBAM and $0.7$ h without CBAM, whose average precision results are 94.1% and 88.5%, respectively. See Table S2, Supporting Information, for details. Since the multidimensional time-series signal has no spatial correlation among different channels, spatial attention compulsively introduces irrelevant information from neighboring channels. This may result in performance degradation of the 2D convolution kernel version with CBAM.

### 3.4. Visualization of the Learned Features by GASF

We adopt the GASF to visualize the output feature maps of GFB and DMB. GASF represents the time-series signal in polar coordinates and encodes the trigonometric summation between each point to identify the correlation within different time intervals. Human eyes are naturally unsuitable for identifying subtle differences in time-series signals through long-term evolution in the wild. Therefore, we transform the time-series signal into a 2D image space by GASF, representing the relative correlation by a superposition of directions concerning the time interval in an intuitive way. Figure 4 provides GASFs of the feature maps after GFB and DMB. The feature maps in the same class always have similar patterns in both feature blocks, and most of the feature maps from different classes are distinguishable. In some cases, the feature maps from the two categories are similar in

### Table 2. Classification results using four feature-based methods and the proposed deep-learning-based method. 95% confidence intervals are included in brackets. The best average results are shown in bold. The $p < 0.05$ indicates that our method significantly improves the compared method (Wilcoxon signed-rank test).

| Network                        | Air  | CGJ  | BYBJX | BYBNX | STJ  | MTWZ | Avg. Prec. | $p$-value |
|-------------------------------|------|------|-------|-------|------|------|-----------|-----------|
| Hand-crafted features-10 + PCA + LDA | 1.000 | 0.286 | 0.556 | 1.000 | 0.333 | 0.444 | 0.603 [0.570, 0.636] | <0.001 |
| Hand-crafted features-20 + PCA + LDA | 1.000 | 0.176 | 1.000 | 0.429 | 0.000 | 0.375 | 0.497 [0.463, 0.530] | <0.001 |
| Data augmentation + PCA + LDA      | 1.000 | 0.100 | 0.750 | 0.833 | 0.625 | 0.571 | 0.797 [0.769, 0.823] | <0.001 |
| Hand-crafted features-10 + PCA + SVM | 1.000 | 0.400 | 0.500 | 1.000 | 0.750 | 0.500 | 0.692 [0.660, 0.722] | <0.001 |
| Hand-crafted features-20 + PCA + SVM | 1.000 | 0.231 | 1.000 | 0.214 | 0.200 | 0.429 | 0.512 [0.478, 0.546] | <0.001 |
| Data augmentation + PCA + SVM      | 1.000 | 0.037 | 1.000 | 0.700 | 0.571 | 0.714 | 0.807 [0.778, 0.831] | <0.001 |
| 1D ResNet-300 + GFB + DMB (AFOA) | 1.000 | 0.852 | 0.742 | 0.789 | 0.527 | 0.664 | 0.762 [0.732, 0.789] | <0.001 |
| 1D ResNet-400 + GFB + DMB (AFOA) | 1.000 | 0.960 | 0.794 | 0.515 | 0.494 | 1.000 | 0.794 [0.765, 0.820] | <0.001 |
| 1D ResNet-500 + GFB + DMB (AFOA) | 1.000 | 1.000 | 0.789 | 0.689 | 0.580 | 0.771 | 0.805 [0.776, 0.830] | <0.001 |
| 1D ResNet-600 + GFB + DMB (AFOA) | 1.000 | 0.940 | 0.990 | 0.777 | 0.704 | 0.690 | 0.850 [0.824, 0.873] | <0.001 |
| 1D ResNet-779 + GFB + DMB (AFOA) | 1.000 | 0.778 | 0.993 | 0.874 | 1.000 | 1.000 | 0.941 [0.922, 0.955] | *<base>*

*Avg. Prec.: average precision, LDA: linear discriminant analysis, PCA: principal component analysis, SVM: support vector machines AFOA: all-feature olfactory algorithm, CGJ: ChunGuJiu, BYBJX: BaiYunBianJianXiang, BYBNX: BaiYunBianNongXiang, STJ: SiTeJiu, and MTWZ: MaoTaiWangZi, Hand-crafted features-10: 10 hand-crafted features calculated during the whole sampling phase, Hand-crafted features-20: 20 hand-crafted features separately calculated during the adsorption and desorption phases. The number after ResNet- indicates the length of the input data.
DMB but distinguishable in GFB; e.g., sample one of BYBJX compares with sample four of STJ. Both the visualization and classification performance results indicate the effectiveness of the two complementary blocks.

For some Chinese liquors, there is not only one pattern in the GASFs, such as for CGJ and MTWZ, indicating that the MOS sensor array has redundant distinguishability for instance classification in one category of Chinese liquor.

3.5. Visualization of Feature Vectors from Different Layers by t-SNE

Feature maps from different layers are high-dimensional time-series signals. In the previous section, we transform them into a 2D image. However, it takes time to build up the right intuition. In 2008, van der Maaten and Hinton proposed a method called t-distributed stochastic neighbor embedding (t-SNE)\(^{[50]}\) for
expanding high-dimensional data, which is widespread in the field of machine learning. t-SNE visualizes high-dimensional data by giving each data point a location in a 2D or 3D map. In this article, we use this technique to create 2D “maps” from feature vectors of different test samples with hundreds of dimensions. In this 2D plane, we can see all the test samples from the different Chinese liquors in clusters. The algorithm is nonlinear and adapts to the underlying data to perform different transformations in different regions. Therefore, the t-SNE plot does a much better job of revealing the natural classes in the data than any linear projection.

Figure 4b–e plots the feature vectors from the different layers of our network, where the color of each dot represents the sample category. From Figure 4b–e, we observe that the intraclass distance at each point gradually decreases, and the interclass distance gradually increases. In Figure 4b, each category is clustered into multiple subcategories. The number of subcategories is always seven, matching the raw number of samples in each category before the data augmentation process (see Figure S1, Supporting Information). This shows that the proposed network has the potential ability to distinguish each instance in the same category, which is similar to the conclusion obtained by QASF.

3.6. The Contributions of Different MOS Sensors

The response curves of an MOS sensor to different odors are particular, and some oxides are more useful for specific recognition tasks. Hence, we investigate the contribution of each sensor to the Chinese liquor recognition task for the guidance of the subsequent material selection and sensor array design. The classification results on the augmented data only for the single sensors and some of their combinations are presented in Figure 5a,b, respectively. Two feature-based methods and our final hybrid model are used for cross-validation. Sensors one (SnO2 QDs) and six (WO3 QDs) outperform the other sensors by a large margin. Furthermore, using the best two sensors is superior to using all the other sensors. In feature-based methods, using sensors one and six is equal to (PCA + LDA: 0.81 vs 0.81) or even better than (PCA + SVM: 0.79 vs 0.74) using the whole sensor array in terms of precision. In general, the deep-learning-based method is no better than feature-based methods in terms of single sensor performance. However, as the number of sensors increases, the performance of our method increases significantly compared to that of feature-based methods. It would appear to suggest that deep-learning-based methods make it easier to learn the patterns from multichannel time-series signals.

3.7. Visualization of the Temporal Attention Weights

To investigate the interpretability of the feature patterns learned by our model, we show the temporal attention weights of the last residual bottleneck in the third backbone stage. Specifically, we interpolate the attention weights to the same length as the input data length to observe the temporal distribution. Figure 6a shows that the beginning of the response rising edge (or descending edge) and the beginning or middle of the recovery period have the greatest influence on distinguishing fine-grained Chinese liquors. The major component of the liquors is ethanol, and the flavors are slightly different. The minor variation in some volatile organic components (VOCs) of these liquors shows a negligible change in the equilibrium state. However, in the response and recovery process, the different VOCs have different chemical kinetic properties, resulting in unique patterns, thus facilitating discrimination. Therefore, the thermokinetic process of the gas sensing reaction, which is usually ignored by traditional intuition-driven methods, is of great significance for odor recognition.

When a single sensor is used to identify the vapors, the temporal attention weights are shown in Figure 6b. For sample 578, the temporal attention weight distributions of sensor one and sensor six are similar to that of the whole sensor array, and the recognition precision is also high. Furthermore, the temporal attention weight distributions of sensor three and sensor five are approximate uniform distributions, but the correct classification results are still obtained. In contrast, the recovery process of sensor two is too slow, and the temporal attention weights of sensor four are mainly concentrated before the vapor enters the measuring chamber, which may be the reason for the incorrect predictions of sensor two and sensor four.

3.8. Comparison of Different Response Curve Modes

To investigate the effects of response curve types of the input data on the model performance, we compare five different response curve modes. As listed in Table S3, Supporting Information, the 3p3n type achieves the best performance in all 6-channel input modes, yielding increments of 11.3% (0.899 vs 0.808), 18.1% (0.899 vs 0.761), 14.1% (0.899 vs 0.788) precision in contrast with 5n1p, 6p, and 6n types, respectively. Interestingly, as shown in Figure S9–S13, Supporting Information, the n-type response curve has more distinguishable features in the equilibrium process, while the p-type response curve has a more suitable representation in the recovery process for liquor identification. The results suggest that the diversity of sensor types and representations may improve the distinguishability of the original input features, thus improving the model performance. Following this indication, we further train a model with 12-channel input, in which each sensor has both n-type and p-type channels. The 6p6n type improves the performance of our method from 0.899 to 0.936 precision compared with the 3p3n input type.

4. Discussion

Our deep-learning-based approach simultaneously simulates the structure and function of the olfactory system, as depicted in Figure 1. Structurally, sensor arrays with different MOS sensors simulate different types of ORCs. The artificial neuron network simulates the olfactory bulb, the olfactory cortex in the brain, and the complex connections between them. Although each kind of metal–oxide material has different responses to odorant molecules, the selectivity of individual metal oxides is low. In addition, due to the limited quantity, volume, and power consumption and the low specificity of semiconductor sensors, as well as the weak feature extraction ability, traditional e-noses have poor odor recognition performance. Therefore, we use a multichannel
nanostructured material to form a sensor array similar to 400 different ORCs in the human olfaction system. Although the variety and number of units in the sensor array are far from those of human ORCs, the experiments show that the performance of e-noses is gradually improved with the increase in sensor units. Functionally, a deep neural network attempts to learn high-level semantic features from massive data. A similar process occurs in human olfaction in which olfactory information is transmitted from olfactory receptors to the glomeruli and finally to the brain. The first convolutional layer receives response curves from the sensor array and detects similar low-level features. Since several sensor units are activated due to the different chemical properties of the odorants, several high-level patterns from intermediate layers are activated. The output of the intermediate layers is then sent to a higher convolutional layer, where the combination of high-level patterns encodes the odorant features. Finally, the classifier puts the activation pattern pieces back together to perceive and identify the odorant.

It should be mentioned that Chinese liquors are usually distilled from fermented sorghum or rice. The contents are very complex, primarily including ethanol and various kinds of esters. Each brand has a unique fragrance by the making process, infusing alcohol with other ingredients, such as herbs and spices (See Table S1 and Figure S2, Supporting Information). Even experienced customers cannot distinguish each other well, which further illustrates the difficulty of the task.

Traditional feature-based methods inspired by domain knowledge mainly contain equilibrium statuses such as the sensor response and resistance in the air and gas statuses. Such hand-crafted features reflect the thermodynamic properties but usually neglect the thermokinetic process of the gas sensing reaction. The essence of chemical reactions is still the collision of atoms. Along with the breaking and formation of chemical bonds, electrons are transferred from one atom to another (i.e., redistribution of the charges). Obviously, the number and efficiency of electron transfer are related to the structure and vibration of the gas molecules and affect the thermokinetics of gas reactions. The visualized dynamics of temporal attention weights support our argument. Due to the different thermokinetic properties of the different VOCs in the Chinese liquors,

**Figure 5.** Ablation study of different MOS sensors on the Chinese liquor dataset. a) Precision classification results of the single sensors. b) Precision classification results of some combinations of single sensors. Sensor 1: SnO2 QDs, Sensor 2: SnO2 nanowires, Sensor 3: SnO2 FSP, Sensor 4: In2O3 QDs, Sensor 5: NiO NPs, Sensor 6: WO3 QDs. PCA: principal component analysis, SVM: support vector machine, LDA: linear discriminant analysis. The error bars represent 95% confidence intervals.
the response and recovery process contribute more to the discrimination results. This is why our AFOA method has better performance than traditional algorithms (Table 2).

Furthermore, the desorption phase contributed more to the classification (Figure 6). All the sensors reached saturation when responding to liquors (Figure 2c). This is because the concentration of the ethanol in the headspace was very high. The sensor's response to other vapors of the liquors was concealed and overlapped, which hinder the capability of further detecting differences in the samples in the adsorption phase. In the desorption phase, the absorbed vapors are desorbed to the ambient air at different speeds related to the chemical kinetics process. The concentrations of absorbed vapors were much lower than that in the headspace, resulting in a relatively low desorption rate.

**Figure 6.** Visualization of the temporal attention weights. a) The temporal attention weights of different samples using all six sensors. b) The temporal attention weights of different sensors in sample 578 of BYBNX. The shaded area indicates the time period in which the target vapor was in the measurement chamber. CGJ: ChunGuJiu, BYBJX: BaiYunBianjianXiang, BYBNX: BaiYunBianNongXiang, STJ: SiTeJiu, and MTWZ: MaoTaiWangZi.
Thus, in the 1-minute desorption phase, the difference between various absorbed vapors is revealed, which facilitates the discrimination.

In the practical environment, the temperature and humidity always vary each day, and the vapor concentration of the sample is hard to maintain at a certain value due to different operators. The algorithm should focus on the intrinsic features of the samples and eliminate the effect of the surrounding environments. Based on this idea, the ambient environment was not controlled, and the operators were not well trained (see 5.2. Acquisition system). Compared to the traditional feature-based methods, our AFOA has a better performance in discriminating different Chinese liquors, which indicates that our algorithm can extract the intrinsic features of the odors.

To investigate the contribution and importance of each sensor unit to the Chinese liquor recognition task, we perform an ablation study on different sensor units and their combinations. The experimental results are consistent with our expectation that more sensor units bring better recognition performance. Complementary sensor units can provide a richer set of features leading to more fine-grained distinguishability. With the change in sensor unit magnitude order, the performance of the e-nose will undergo tremendous changes. At present, the main problem of the sensing hardware is the rational design and controllable preparation of sensor arrays. The ablation study results for different sensor units and visualized attention weights can iteratively bring crucial guiding significance for material selection and odor recognition strategies in application-specific sensing tasks.

The reason why sensors one and six are better than the others can be explained as follows: QDs have a small diameter ($\approx 10$ nm) resulting in a large specific area that enhances gas molecule adsorption.[52] Moreover, the long-chain ligand surrounded by the QD surface also participates in the gas reaction process, which can improve the gas sensing properties. Each QD acted like a mammalian olfactory receptor, while the QD sensor acted like an olfactory bulb. The QD sensor showed excellent sensing performance, which contributed more than other sensors to discrimination.

To increase the scale and the diversity of the sensor response signals, some mathematical functions are used to transform the raw data into a suitable representation, such as reciprocal transformation and power function transformation. This representation enriches the characteristics of the input data in terms of time, channel, and sensor response process which are related to the reaction dynamics underlying the gas molecules’ reception and signal transduction, thereby benefitting both deep-learning-based and multistage feature-based methods. At the physicochemical level, the surface oxygen molecules accept the electrons of the n-type gas sensor and form the depletion layer. Upon exposure to the reducing VOCs, the redox reaction with the absorbed oxygen releases the trapped electrons back to the sensor, which shortens the depletion layer. On the contrary, in the p-type sensor where the majority of the carrier is a hole, the accumulation layer forms and reacts based on a process similar to the oxygen chemisorption.[52] The different conduction type thereby represents different thermokinetic and thermos-dynamic processes, inherently providing more distinguishing features to the algorithm. One can expect that the diverse gas sensors would lead to richer inherent characteristics. While it is still far from reaching a universal design strategy, our study suggests that the gas sensor array containing both n-type and p-type sensors is beneficial to the e-nose.

Taken together, we propose a novel network architecture specifically designed for MOS array-based odor recognition that leverages a CNN and an RNN[53] that outperforms traditional two-stage methods by a large margin. Our approach is superior to any individual CNN or RNN network for temporal feature extraction. Second, our method uses 1D convolution kernels to process the raw data of the MOS array for its natural properties, significantly improving the efficiency and speed of calculation. Third, a data augmentation process is designed, especially for the time-series signal, to improve the distinguishability of the original input data for the subsequent representational learning. In short, the hidden features extracted from the all-feature extraction method exhibit the different parameters of the reactions between gas and sensing materials so that different odors can be distinguished as specific patterns.

Due to the generality of our method, we can widely apply it to other time-series signal modeling tasks, such as music analysis, underwater object detection, and gesture recognition. However, the drawback of AFOA is that it requires a full response and recovery curves for discrimination, which takes several minutes, thus limiting its application for fast recognition. Our future work will be dedicated to rapid real-time odor detection and recognition in the wild with a low sampling rate by enhancing the sampling method and adjusting the algorithm, which is a much more challenging task due to the interference of irrelevant odors, ambient conditions (various humidity and temperature changes) and computational power constraints.

5. Conclusion

In conclusion, our work presents an e-nose simply combined with six MOS gas sensors and a novel deep-learning-based algorithm. The tailored all-feature extraction method and the proposed data augmentation method offer a superior advantage in complex odor discrimination over feature-based methods, especially in a complex circumstance (uncontrolled temperature and humidity). We also establish a deep-learning-enabled e-nose codesign paradigm that automatically learns target-specific features to iteratively optimize sensor materials and sampling strategies. Further illustrating the features can lead to a way to determine the intrinsic properties of a gas sensing mechanism and guide the precise design of a high-performance e-nose. However, the “olfactory codes” of different molecules are still worthy of further study. In the future, the high integration of the gas sensor array and the intelligent algorithm will make the e-nose surpass the biological nose.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

C.F., H.-Y. L. contributed equally to this work. This work was supported by the National Natural Science Foundation of China (61922032). The authors
thank the Program for HUST Academic Frontier Youth Team (2018QYTD06) and the Innovation Fund of WNLO. The authors thank the Analytical and Testing Center of HUST for the characterization support.

Conflict of Interest
The authors declare no conflict of interest.

Author Contributions
C.F., H.Y.L., H.L., J.T., and X.B. proposed and designed the study. C.F. developed odor recognition algorithms under X.B.’s guidance. L.L., H.S., and H.Y.L. collected the data. C.F., H.Y.L., and H.L. interpreted the data. C.F. and H.Y.L. wrote the manuscript under H.L. and X.B.’s guidance. All authors read and approved the final manuscript.

Data Availability Statement
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords
all-feature extraction, deep learning, electronic nose, odor recognition, sensor array

Received: March 18, 2022
Revised: April 9, 2022
Published online: May 12, 2022

[1] L. Buck, R. Axel, Cell 1991, 65, 175.
[2] C. Bushdid, M. Magnano, L. Vosshall, A. Keller, Science 2014, 343, 1370.
[3] K. Persaud, G. Dodd, Nature 1982, 299, 352.
[4] S.-Y. Jeong, J.-S. Kim, J. Lee, Adv. Mater. 2020, 32, 2002075.
[5] M. Kaneyasu, A. Ikegami, H. Arima, S. Iwanaga, IEEE Trans. Compon., Hybrids, Manuf. Technol. 1987, 10 267.
[6] T. Pearce, J. Gardner, S. Friel, in Systems Man and Cybernetics Conf. (SMC), IEEE, Piscataway, NJ 1993, pp. 165–170.
[7] E. L. Hines, J. W. Gardner, Sens. Actuators, B 1994, 19 661.
[8] J. W. Gardner, T. C. Pearce, S. Friel, P. N. Bartlett, N. Blair, Sens. Actuators, B 1994, 18 240.
[9] F. Röck, N. Barsan, U. Weimar, Chem. Rev. 2008, 108, 705.
[10] A. Loutfi, S. Coradeschi, G. Mani, P. Shankar, J. Rayappan, J. Food Eng. 2015, 144, 103.
[11] A. Wilson, M. Baietto, Sensors 2009, 9, 5099.
[12] J. Gutiérrez, M. Horrillo, Talanta 2014, 124, 95.
[13] N. Barsan, D. Koziej, U. Weimar, , Sens. Actuators, B 2007, 121, 18.
[14] Q. Zhang, C. Xie, S. Zhang, A. Wang, B. Zhu, L. Wang, Z. Yang, , Sens. Actuators, B 2005, 110, 370.
[15] Q. Zhou, S. Zhang, Y. Li, C. Xie, H. Li, X. Ding, Sens. Actuators, B 2011, 160, 483.
[16] H. Qin, D. Huo, L. Zhang, L. Yang, S. Zhang, M. Yang, C. Shen, C. Hou, Food Res. Int. 2012, 45, 45.
[17] H.-R. Hou, Y.-J. Liu, Q.-H. Meng, in 8th Annual Int. Conf. on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) IEEE, Tianjin, China 2018, pp. 297–301.
[18] P.-F. Qi, Q.-H. Meng, Y.-Q. Jing, Y.-J. Liu, M. Zeng, IEEE Sens. J. 2017, 17, 4689.
[19] Y.-J. Liu, M. Zeng, Q.-H. Meng, Rev. Sci. Instrum. 2019, 90, 025001.
[20] X. Wu, J. Zhu, B. Wu, C. Zhao, J. Sun, C. Dai, Foods 2019, 8.
[21] Y. Jing, Q. Meng, P. Qi, M. Zeng, W. Li, S. Ma, Rev. Sci. Instrum. 2014, 85, 055004.
[22] Q. Li, Y. Gu, N.-F. Wang, IEEE Sens. J. 2017, 17, 1788.
[23] H.-R. Hou, Q.-H. Meng, X. Zhang, in 24th Int. Conf. on Pattern Recognition (ICPR) IEEE, Beijing, China 2018, pp. 237–241.
[24] M. Liu, M. Wang, J. Wang, D. Li, Sens. Actuators, B 2013, 177, 970.
[25] S. Qiu, J. Wang, C. Tang, D. Du, J. Food Eng. 2015, 166, 193.
[26] Q. Li, Y. Gu, J. Jia, Sensors 2017, 17, 272.
[27] M. Laska, P. Teubner, Chem. Sens. Actuators, B 1999, 24 161.
[28] W. H. Overman, L. Boettcher, L. Watterson, K. Walsh, Behav. Brain Res. 2011, 218 64.
[29] H. N. Schifferstein, K. S. Talke, D.-J. Oudshoorn, Chemosens. Percept. 2011, 4 55.
[30] I. Goodfellow, Y. Bengio, A. Courville, Nature 2015, 521, 436.
[31] G. Wei, G. Li, J. Zhao, A. He, Sensors 2019, 19, 217.
[32] L. Han, C. Yu, K. Xiao, X. Zhao, Sensors 2019, 19, 1960.
[33] Q. Wang, T. Xie, S. Wang, in Chinese Automation Congress (CAC) IEEE, Xi’an, China 2018, pp. 3486–3491.
[34] A. Ip, S. Thon, S. Hoogland, O. Voznyy, D. Zhitomirsky, R. Debnath, L. Levina, L. R. Rollny, G. H. Carey, A. Fischer, K. W. Kemp, I. J. Kramer, Z. Ning, A. Labelle, K. W. Chou, A. Ammassian, E. Sargent, Nat. Nanotechnol. 2012, 7, 577.
[35] Z. Ning, O. Voznyy, J. Pan, S. Hoogland, V. Adinolfi, J. Xu, M. Li, A. R. Kirmani, J.-P. Sun, J. C. Minor, K. W. Kemp, H. Dong, L. R. Rollny, A. Labelle, G. H. Carey, B. Sutherland, I. Hill, A. Ammassian, H. Liu, J. Tang, O. Bakr, E. Sargent, Nat. Mater. 2014, 13, 822.
[36] K. He, X. Zhang, S. Ren, J. Sun, in Conf. on Computer Vision and Pattern Recognition (CVPR) IEEE, Las Vegas, NV, USA 2016, pp. 770–778.
[37] S. Ioffe, C. Szegedy, in International conference on machine learning (ICML) JMLR.org, Lille, France 2015, pp. 448–456.
[38] V. Nair, G. E. Hinton, in International conference on machine learning (ICML) JMLR.org, Hafa, Israel 2010, pp. 807–814.
[39] S. Woo, J. Park, J.-Y. Lee, I.-S. Kweon, in European Conference on Computer Vision (ECCV) Springer, Munich, Germany 2018, pp. 3–19.
[40] S. Hochreiter, J. Schmidhuber, Neural Comput. 1997, 9, 1735.
[41] W. Zaremba, I. Sutskever, O. Vinyals, ArXiv 2014, abs/1409.2429.
[42] A. Graves, in Springer 2012.
[43] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, J. Mach. Learn. Res. 2014, 15, 1929.
[44] D. P. Kingma, J. Ba, in ICLR ICLR, San Diego, CA USA 2015.
[45] A. Skarysz, Y. Alkalafih, K. Darnley, M. Eddleston, Y. Hu, D. McLaren, W. Nailon, D. Salmon, M. Sykora, C. P. Thomas, A. Soltoggio, in Int. Joint Conf. on Neural Networks (IJCNN) IEEE, Rio de Janeiro, Brazil 2018, pp. 1–8.
[46] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, Nature 2020, 585 357.
[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Mach. Learn. Res. 2011, 12, 2825.
[48] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, Nat. Methods 2020, 17 261.
[49] Z. Wang, T. Oates, in IJCAI IJCAI, Buenos Aires, Argentina 2015, pp. 3939–3945.
[50] L. V. D. Maaten, G. E. Hinton, Adv. Intell. Syst. 2022, 4, 2200074 2200074 (15 of 15) © 2022 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH