applied sciences

MDPI

Communication

A Sensor Drift Compensation Method with a Masked Autoencoder Module

Seokjoon Kwon [1], Jae-Hyeon Park [2], Hee-Deok Jang [1], Hyunwoo Nam [3] and Dong Eui Chang [1,*]

[1] School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea; jun115533@kaist.ac.kr (S.K.); jhd6844@kaist.ac.kr (H.-D.J.)
[2] Robot Innovation Group, Samsung Display Co., Yongin-si 17113, Republic of Korea; jaehyeon.park53@samsung.com
[3] Chem-Bio Technology Center, Agency for Defense Development, Daejeon 34186, Republic of Korea; hyunwoonam@add.re.kr
* Correspondence: dechang@kaist.ac.kr

Abstract: Deep learning algorithms are widely used for pattern recognition in electronic noses, which are sensor arrays for gas mixtures. One of the challenges of using electronic noses is sensor drift, which can degrade the accuracy of the system over time, even if it is initially trained to accurately estimate concentrations from sensor data. In this paper, an effective drift compensation method is introduced that adds sensor drift information during training of a neural network that estimates gas concentrations. This is achieved by concatenating a calibration feature vector with sensor data and using this as an input to the neural network. The calibration feature vector is generated via a masked-autoencoder-based feature extractor trained with transfer samples, and acts as a prompt to convey sensor drift information. Our method is tested on a 3-year gas sensor array drift dataset, showing that a neural network using our method performs better than other models, including a network with additional fine tuning, demonstrating that our method is efficient at compensating for sensor drift. In this study, the effectiveness of using prompts for network training is confirmed, which better compensates for drifts in new sensor signals than network fine-tuning.

Keywords: sensor calibration; sensor drift; deep learning; masked autoencoder; representation learning; prompt-based learning

Citation: Kwon, S.; Park, J.-H.; Jang, H.-D.; Nam, H.; Chang, D.E. A Sensor Drift Compensation Method with a Masked Autoencoder Module. *Appl. Sci.* **2024**, *14*, 2604. https://doi.org/10.3390/app14062604

Academic Editor: Christos Bouras

Received: 26 January 2024
Revised: 7 March 2024
Accepted: 11 March 2024
Published: 20 March 2024

1. Introduction

An electronic nose (E-nose) is a system comprising an array of gas or chemical sensors and a pattern recognition module that can identify and measure simple or complex gases [1]. E-noses are increasingly studied and they have been widely used in various fields, such as the food industry, agriculture, healthcare, air pollution monitoring, and security systems [2]. Various sensor applications utilize machine learning algorithms for specific purposes [3], and these algorithms are also used to process multiple time-varying sensor signals generated in E-noses to quantify and identify target gases. Initially, E-nose algorithms employed linear techniques, such as principal component analysis and partial least squares regression [2,4,5], or nonlinear techniques such as support vector machines [2,6]. Nowadays, applying deep learning to E-noses has become more popular due to its superior ability to handle complex data [7–10].

Although conventional machine learning is highly effective for E-nose applications, its long-term performance is hindered by sensor drift, which is a phenomenon that causes temporary or gradual changes in sensor characteristics due to factors such as aging, contamination, or environmental fluctuations [1]. To address this issue, researchers have developed a method known as component correction, which involves identifying and modeling the drift direction in reference samples and subtracting it from new data [5]. While component correction is typically performed using linear techniques such as principal component

analysis, partial least squares regression, or independent component analysis [11], it does not account for the nonlinear behavior of the drift. A representative approach to compensate for the drift in deep learning is fine-tuning new sensor data, adjusting the weight of some or all neurons in response to new samples [5]. Adaptive drift correction, which is a method that updates a classifier continuously using new samples, has been implemented in various deep learning structures [5]. Fine-tuning has also been carried out for novel deep learning models such as autoencoders [12], restricted Boltzmann machines [13], deep belief networks [14], and augmented convolutional neural networks [15]. Furthermore, online drift compensation methods, which can update trained models with new samples, have been studied. An online domain adaptation extreme learning machine, a combination of an existing domain adaptation extreme learning machine with an online learning theory, was utilized in [16] for gas identification. To reduce the annotation effort, an active learning method was applied to online drift compensation in [17]. This field of online sensor drift compensation has been expanded not only to gas classification but also to gas concentration estimations, and this research trend regarding sensor drift compensation via fine-tuning methods is still ongoing [18]. Nonetheless, the fine-tuning process can be time-consuming and complex, which may be problematic given the limited onboard memory in sensor systems [5,15], and there are still many unsolved challenges [18].

Meanwhile, in the field of natural language processing, the concept of prompt-based learning has been actively studied as a substitute for fine-tuning [19–21]. A prompt is a snippet of context information that is added to the input training data during the training phase. During the training phase, a model learns to associate the prompts with the corresponding inputs and outputs. This method has been shown to be effective for large language models when used for different tasks without fine-tuning [19,21,22]. Prompts can be generated by humans or by the model, and the latter is called auto prompting. In this case, a model automatically produces an adequate prompt corresponding to data [23,24]. The prompts can be classified as discrete prompts or continuous prompts based on their form. Discrete prompts are natural language, while continuous prompts are embedding vectors which function as a prompt [25,26]. We use the concept of prompt-based learning to design an effective drift compensation method for a deep learning algorithm that estimates gas concentrations.

In this paper, we propose a method for training neural networks by augmenting the input sensor data with a prompt which contains sensor drift information. We call the prompt a calibration feature vector, which functions as a continuous prompt generated by a feature embedding module called a calibration feature encoder (CFE) and it is concatenated to input sensor data. In a drifted environment, transfer samples are obtained by collecting sensor signals and subsequently determining the concentration of gases in that environment. The CFE is a masked-autoencoder-based module which is trained with past transfer samples to produce calibration feature vectors for future transfer samples. In contrast to fine-tuning methods, which tune a neural network using new transfer samples from the drifted environment, our method trains a neural network only once using past sensor samples with prompts produced by the CFE. The neural network learns to estimate gas concentrations with sensor data and drift information, which makes it robust to future sensor drift without fine-tuning while simultaneously improving the performance. Our method is capable of learning nonlinear drift behavior, does not require neural network fine-tuning, and shows superior performance compared to other models, including a neural network with fine-tuning. The main contributions of this paper are as follows:

- We propose a training method for deep learning models that estimate gas concentrations by concatenating a prompt which contains sensor drift information with input sensor data. By training the model with the prompt, the model is robust to sensor drift without fine-tuning.
- We utilize a masked-autoencoder-based CFE for the effective feature extraction of sensor drift information. The experimental results demonstrate that the deep learn-

ing model using a CFE outperforms the control group, indicating that the CFE can effectively generate prompts containing sensor drift information.

The remainder of this paper is as follows. Section 2 explains the experimental data, overall concept, and the proposed methodology. The experimental settings and results are discussed in Section 3. Finally, Section 4 provides the conclusions and describes future works.

2. Materials and Methods

2.1. Dataset

For our experiments, we utilized the publicly available dataset that was constructed by Vergara et al. [27], which is a well-known dataset used in drift compensation research [7,12,14,15,28–30]. This dataset comprises 13,910 measurements gathered from an E-nose device with 16 chemical sensors that were exposed to the 6 gases of ethanol, ethylene, ammonia, acetaldehyde, acetone, and toluene at various concentrations. The data were accumulated over the course of three years and were divided into 10 batches, which were chronologically arranged as shown in Table 1. Each sample in the dataset is represented by a 128-dimensional feature vector, which was extracted from the E-nose signal along with the corresponding gas type and concentration. A detailed explanation of the dataset and the feature vector can be found in the paper by Vergara et al. [27].

Table 1. Details about the dataset by Vergara et al. [27].

Batch ID (Months ID)	Number of Samples					
	Ethanol	Ethylene	Ammonia	Acetaldehyde	Acetone	Toluene
Batch 1 (Months 1–2)	90	98	83	30	70	74
Batch 2 (Months 3–10)	164	334	100	109	532	5
Batch 3 (Months 11–13)	365	490	216	240	275	0
Batch 4 (Months 14–15)	64	43	12	30	12	0
Batch 5 (Months 16)	28	40	20	46	63	0
Batch 6 (Months 17–20)	514	574	110	29	606	467
Batch 7 (Months 21)	649	662	360	744	630	568
Batch 8 (Months 22–23)	30	30	40	33	143	18
Batch 9 (Months 24–30)	61	55	100	75	78	101
Batch 10 (Months 36)	600	600	600	600	600	600

We categorized the dataset into three groups based on the elapsed time. Dataset-A includes samples from months 1 to 20 which represent past samples, and dataset-B and dataset-C include samples from months 21 to 23 and months 24 to 36, respectively. Dataset-B and -C were used to analyze the drift compensation ability in addition to fine-tuning in later experiments for comparison purposes.

2.2. Overall Concept

In this section, we discuss the fundamental principles of the CFE and how it can be implemented. The steady-state response of gas sensors can be written in an abstract form as follows [31]:

$$R = f(R_0, K, T, C),$$

where $R \in \mathbb{R}^{n_s}$ is the resistance of the sensor chemicals in response to the gas mixture, n_s is the number of gas sensors, f is a function derived from chemistry, $R_0 \in \mathbb{R}^{n_s}$ is the base

resistance of the sensor chemicals, $K \in \mathbb{R}^{n_k}$ is the sensitivity coefficient, n_k is the number of coefficients, $T \in \mathbb{R}$ is the temperature, $C \in \mathbb{R}^{n_c}$ is the concentration of gases in the mixture, and n_c is the number of gases. Since the form of the function f is complex and K is unknown, we use pattern recognition algorithms to estimate the gas concentration C given measured values of R, R_0, and T. Even if we train a pattern recognition algorithm with samples, the values of R_0 and K change after a long time due to sensor drift, which is the rationale behind drift compensation methods.

Transfer samples are pairs of a measured sensor signal and a concentration which are taken some time after the first training of the E-nose's pattern recognition algorithm [12]. These samples are collected using the same E-nose over a small time interval in which the drift of the E-nose is negligible. The concept behind the CFE is based on the fact that transfer samples carry information on f, R_0, and K. For instance, assuming that we know the form of f and the temperature T, we can obtain the values of R_0 and K from the transfer samples using optimization or equation-solving techniques, even in the presence of sensor drift. By utilizing transfer samples to train the CFE, sensor drift features are extracted and a calibration feature vector is generated that contains sensor drift information. In normal neural networks for concentration estimation [8,32], the sensor signal values are input and the corresponding concentration is predicted, as shown in Figure 1a. To address sensor drift, network fine-tuning is performed by using new transfer samples to adapt to the drifted R_0 and K values. In our proposed method, rather than tuning the parameters with new transfer samples, we train the neural network with sensor data concatenated with a calibration feature vector produced from the CFE, which makes the network robust to sensor drift without fine-tuning.

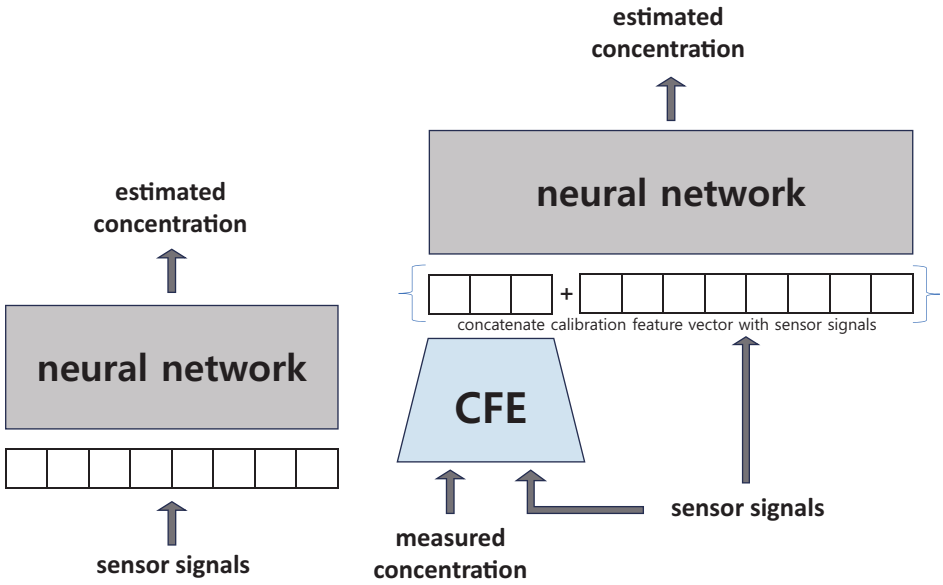


Figure 1. (**a**) Conventional neural network (**b**) Neural network with a prompt that is the calibration feature vector generated from CFE, which is concatenated with a sensor signal.

2.3. Proposed Method

In the proposed method, a deep learning model is trained to estimate gas concentrations by concatenating a calibration feature vector which contains sensor drift information to input sensor data. By training the model with a calibration feature vector, it learns to estimate gas concentrations by correlating the input sensor data and sensor drift information, making the model robust to sensor drift without fine-tuning. This methodology of prompt learning, i.e., training a neural network with context information, has shown to be effective when learning a robust model without fine-tuning for new data [19–21]. To extract good sensor drift features, we design a masked-autoencoder-based encoder, the CFE, which is

trained with past transfer samples to generate calibration feature vectors. The main aim of masked autoencoders in computer vision tasks is to remove random patches in an input image by masking and training an autoencoder model to reconstruct the original image. Masked autoencoders have been proven to produce complex, holistic reconstructions that lead to a good performance in many computer vision tasks [33]. We applied this idea to design a CFE to extract calibration feature vectors, as illustrated in Figure 2. After the CFE is trained, we trained a neural network that estimates gas concentrations only once with past sensor data concatenated with a calibration feature vector produced by the CFE. The CFE's procedure is illustrated in Figure 1b.

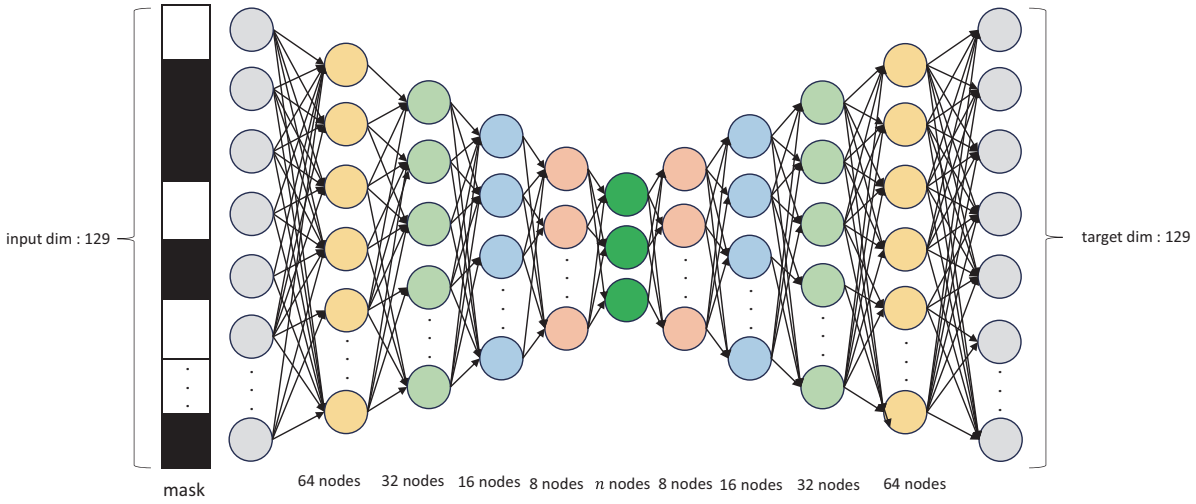


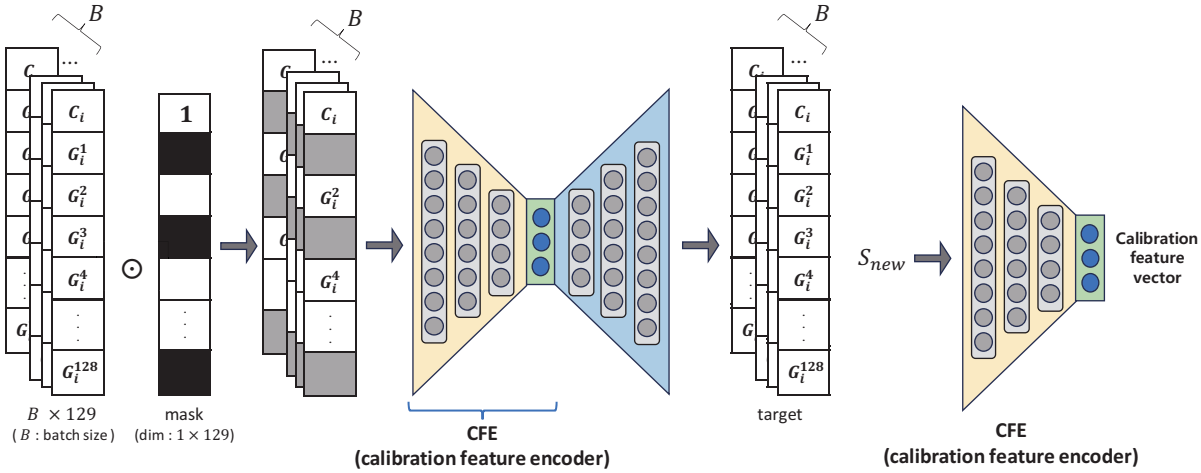
Figure 2. Architecture of the designed masked autoencoder. The dimension of the input data is 129 since it is comprised of a 128 dimension sensor signal and a corresponding concentration value.

The input data for the CFE are transfer samples which are composed of 128-dimensional sensor signals and corresponding concentration values. The transfer sample in the i-th data is denoted as S_i, which is $\{C_i, G_i^1, G_i^2, G_i^3, \ldots, G_i^k\}$, where G_i^j denotes the j-th value of sensor signals in the i-th data and C_i denotes the concentration value in the i-th data. In our case, k is 128 since the sensor signal sample from the E-nose is represented by a 128-dimensional feature vector. The mask is a 129-dimensional vector consisting of 0s and 1s, which is then multiplied by the Hadamard product with each batch of training data. For the input data, 75% of the gas sensor data are masked out and this mask is generated randomly for each batch while the concentration value remains the same. The target is the same data as the original input data before masking, and the masked autoencoder is trained to reconstruct the target from the masked data. Thus, the objective function to be minimized is the reconstruction error between the input data and the reconstructed data as follows:

$$L_{MSE} = \frac{1}{B}\sum_{i=1}^{B} \|S_i - \hat{S}_i\|^2,$$

where $S_i \in \mathbb{R}^{129}$ and $\hat{S}_i \in \mathbb{R}^{129}$ are the i-th transfer sample and the i-th reconstructed output in each batch, respectively, and B is the number of data included in the batch. When the training is complete, the encoder part is extracted and used as a module to produce calibration feature vectors from new transfer samples. This procedure is illustrated in Figure 3. The designed masked autoencoder includes 5 layers with 129, 64, 32, 16, and 8 nodes sequentially at the encoder and 5 layers with 8, 16, 32, 64, and 129 nodes sequentially at the decoder. The latent space dimension is n and transfer samples from dataset-A are used for training. The training, validation, and test data are partitioned in a 7:2:1 ratio and all samples are normalized using standard normalization with the mean and standard deviation of the training set. For training, we use the Adam optimizer, the ReLU activation

function, and a learning rate of 0.0001 for 500 epochs, and early stopping is applied when the performance does not improve further on the validation set for 100 subsequent epochs.



(**a**) (**b**)

Figure 3. (**a**) Training phase of CFE. A masked autoencoder model is trained with past transfer samples. The j-th value of sensor data in the i-th data is denoted by G_i^j. The value of concentration in the i-th data is denoted by C_i. The batch size is denoted by B. The symbol $\odot$ denotes the Hadamard product. The first value of the mask is always 1 because the concentration value is kept unchanged. (**b**) CFE that is extracted from the trained masked autoencoder model at the training phase in the left figure. A new transfer sample which is a pair of new sensor signals and corresponding concentration information is denoted by S_{new}.

After the CFE is extracted from the trained masked autoencoder, a gas concentration estimation model is trained with past sensor data concatenated with a calibration feature vector. We designed a multi-layer perceptron (MLP) consisting of several fully connected layers for concentration estimations as a representative model for gas concentration estimation as multi-layer perceptrons are commonly used in gas identification and gas concentration estimation tasks from sensor values [8,34,35]. The MLP is trained using sensor data from dataset-A concatenated with a calibration feature vector generated from samples of the same dataset. The CFE is not trained further but generates prompts, which means that it just produces extra information for the MLP. The hidden layers of the MLP here include 3 layers with 32 nodes and 4 layers with 16 nodes, and the input dimension is $128 + n$, where n is the latent space dimension of the CFE. A dropout layer of 0.3 is applied to the last hidden layer to prevent overfitting, and the loss function, the optimizer, the activation function, and the learning rate are the same as those used for training the CFE. The loss function is as follows:

$$L_{MSE} = \frac{1}{B}\sum_{i=1}^{B}(y_i - \hat{y}_i)^2,$$

where $y_i \in \mathbb{R}$ and $\hat{y}_i \in \mathbb{R}$ are the i-th ground truth concentration value and the i-th estimated concentration value in each batch, respectively, and B is the number of data included in the batch. Early stopping is also applied while training in the same way as it is applied in training the CFE. For training, the MLP uses samples from dataset-A, and the partition ratio of the training, validation, and test data and the normalization method are the same as those for the dataset used for the CFE. We denote this MLP trained with calibration feature vectors as MLP-CFE.

3. Experiment and Results

3.1. Experiment Setting

We designed five different models to compare with the MLP-CFE. The first is the MLP-normal model, which is an MLP consisting of an input layer, hidden layers, and an output layer. Only sensor values are used as input; the hidden layers include 12 layers with 32 nodes, 9 layers with 16 nodes, and a dropout layer. The output layer has one node for predicting the gas concentration. The second model is a partial least squares (PLS) model which is a representative linear technique method. We chose this model since PLS is a popular linear feature extractor [36–38]. We trained the PLS model on each gas and chose the optimal number of principal components based on the accuracy for each gas. The third model is the MLP-PLS, which has the same structure as the MLP-CFE except for the input dimension. As the MLP-PLS model uses a trained PLS model as a feature extractor to compensate for sensor drift, the input dimension is $128 + k$, where k is the number of principal components in the PLS model for each gas, which is 30, 31, 7, 32, 28, and 28 for ethanol, ethylene, ammonia, acetaldehyde, acetone, and toluene, respectively. The fourth model is an MLP-AE which shares the same architecture and utilizes the same training dataset as the MLP-CFE, but the autoencoder is unmasked during training. The fifth model is the MLP-tuned model, which has the same structure as the MLP-normal model but whole layers with transfer samples from new sensor data are fine-tuned. We tested the MLP-CFE and MLP-AE models using an encoder module with different latent space dimensions ($n = 2, 3, 4$, where n denotes the latent space dimension). We used comparable numbers of parameters for the models that include an MLP for the sake of a fair comparison; see Table 2 for the number of parameters used. The MLP-encoder model in the table includes both the MLP-CFE and MLP-AE models, but the number of parameters is the sum of the encoder module parameters and the MLP parameters. The MLP-tuned model has the same number of parameters as the MLP-normal model because they have the same architecture. All algorithms were implemented in Python and Pytorch.

Table 2. The number of parameters of models, where n denotes for latent space dimension of the autoencoder model.

Model	Number of Parameters					
	Ethanol	Ethylene	Ammonia	Acetaldehyde	Acetone	Toluene
MLP-normal	18,465	18,465	18,465	18,465	18,465	18,465
MLP-PLS	19,425	19,457	18,689	19,489	19,361	20,353
MLP-encoder ($n = 2$)	18,747	18,747	18,747	18,747	18,747	18,747
MLP-encoder ($n = 3$)	18,788	18,788	18,788	18,788	18,788	18,788
MLP-encoder ($n = 4$)	18,829	18,829	18,829	18,829	18,829	18,829

All models, except for the MLP-tuned, were trained, validated, and tested using samples from dataset-A, while the training, validation, and test data were partitioned in a 7:2:1 ratio. A total of 10 percent of samples from dataset-B and dataset-C were set aside as the second and third test sets, respectively, to analyze the drift compensation ability of these models. All the test sets were normalized using standard normalization with the training set of dataset-A because the models were trained with this dataset. MLP-tuned was trained through the following three steps. For data from 1–20 months, it was trained, validated, and tested using samples from dataset-A. For data from 21–23 months, it was trained, validated, and tested with samples from dataset-B using the parameters of the model initialized based on the parameters from the previous step. Finally, for data from 24–36 months, MLP-tuned undergoes the same process as that of the second step with samples from dataset-C. The samples at each step are normalized by performing standard normalization of the training sets of each dataset. During this training process, the fine-tuning process is implemented with new transfer samples. Figure 4 shows a schematic diagram of training for MLP-tuned. When training the autoencoder models and PLS, transfer samples are randomly selected from dataset-A.

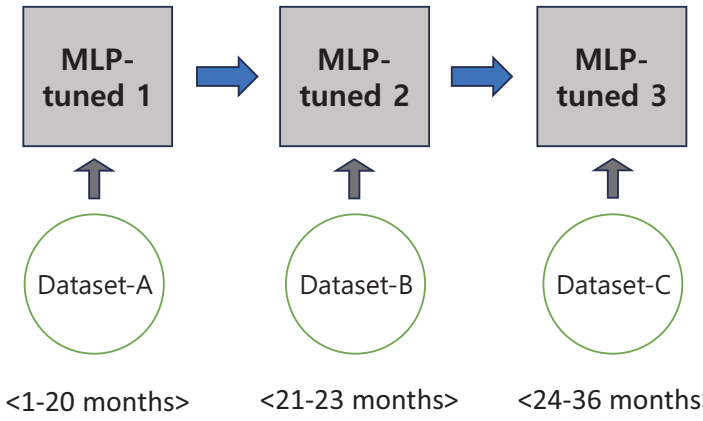


Figure 4. A schematic diagram of training process of MLP-tuned. MLP-tuned undergoes training at each step using the dataset of that step, with the parameters of the model initialized based on the parameters from the previous step.

3.2. Results

We calculated the root mean squared error (RMSE) of concentration estimations for each model with the test set of each gas type listed in Table 1. The resulting RMSEs are plotted in Figure 5, where n denotes the latent space dimension and was set to 3. In Figure 5, the RMSE generally increases with time due to sensor drift. The top three models with the lowest RMSE for the test set of dataset-C are MLP-tuned, MLP-CFE, and MLP-AE for every gas. For the test set of dataset-B, except for ammonia and toluene, the top three models with the lowest RMSEs are MLP-tuned, MLP-CFE, and MLP-AE.

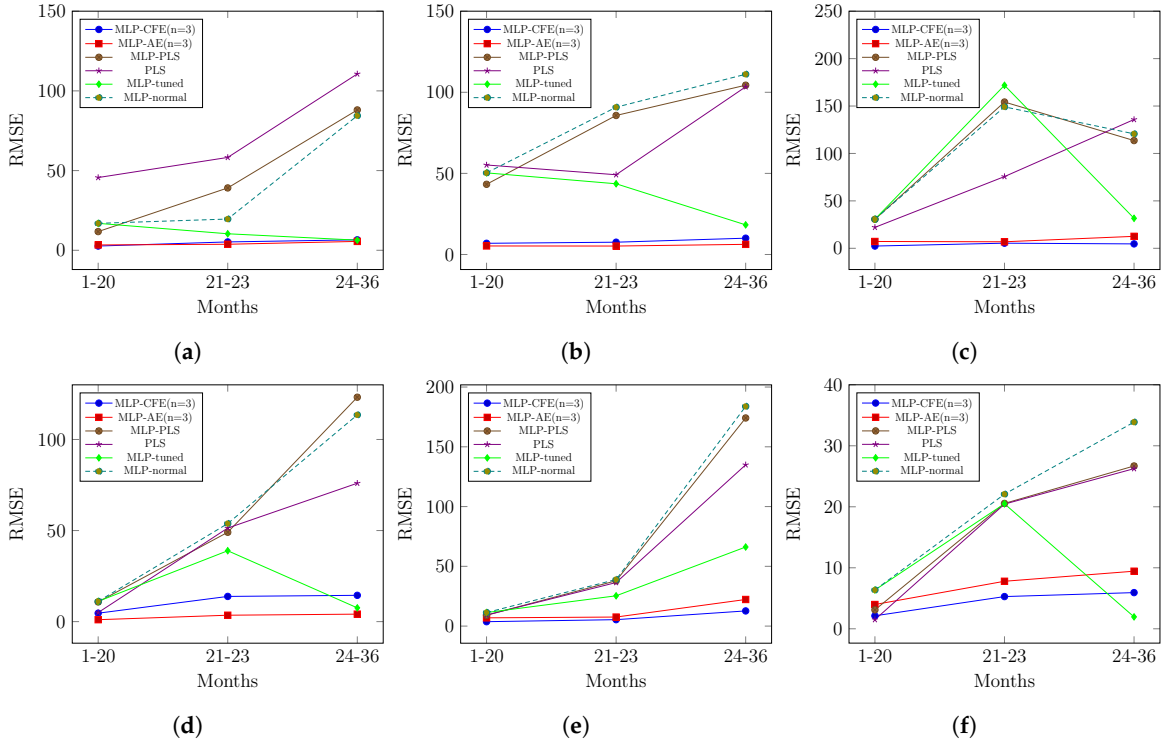
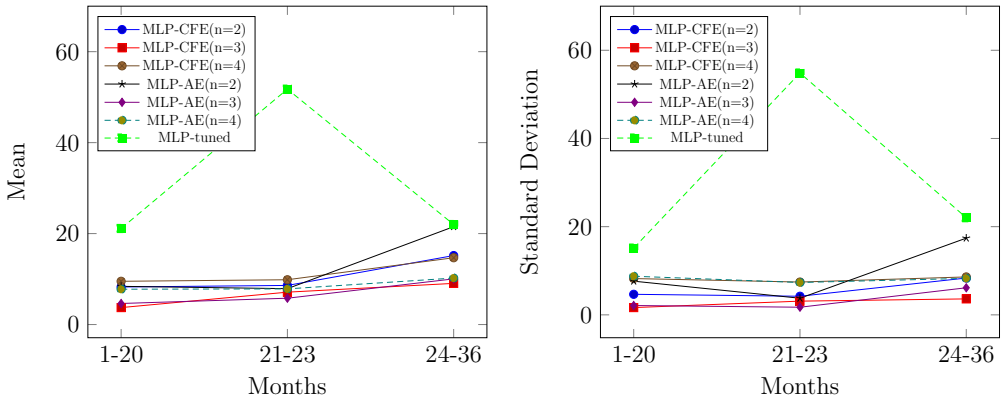


Figure 5. RMSE graphs of (**a**) Ethanol, (**b**) Ethylene, (**c**) Ammonia, (**d**) Acetaldehyde, (**e**) Acetone, and (**f**) Toluene for each models.

For the six gases, the mean RMSE and the standard deviation of the RMSE, which represent the average accuracy of gas concentration estimations and robustness to gas type, respectively, were compared for MLP-tuned, MLP-CFE, and MLP-AE (Figure 6 and Table 3). For this comparison, latent layer dimensions of two, three, and four are also used for MLP-

CFE and MLP-AE. MLP-CFE ($n = 3$) shows the lowest mean RMSE on dataset-A and -C test samples. On dataset-B test samples, it shows the second lowest mean RMSE, with a small difference of 1.33 compared to MLP-AE ($n = 3$). For the standard deviation of the RMSE, MLP-CFE ($n = 3$) shows the lowest RMSE on dataset-A and -C test samples, while for dataset-B test samples, it shows the second lowest RMSE, with a small difference of 1.38 compared to MLP-AE ($n = 3$). When the mean RMSEs for 0–20 months, 21–23 months, and 24–36 months are averaged for each of the models, MLP-CFE ($n = 3$) exhibits the lowest value of 6.63. In addition, when the standard deviations of the RMSE for these three periods were averaged in the same way, MLP-CFE ($n = 3$) exhibits the lowest value of 2.79. Notably, the MLP-CFE ($n = 3$) model, which is our proposed model, exhibits a superior performance over the other models with regard to the mean and the standard deviation of the RMSE for all test sets. In particular, for the dataset-C test set, the best performing model is MLP-CFE ($n = 3$), which means that our method for compensating sensor drift is efficient for long-term sensor drift and also robust to different types of gases.



(**a**) (**b**)

Figure 6. (**a**) Mean of RMSE of 6 gases for each model (**b**) Standard deviation of RMSE of 6 gases for each model.

Table 3. The mean of RMSE and the standard deviation of RMSE for each model. SD denotes for standard deviation. n for latent space dimension.

	0–20 Months		**21–23 Months**		**24–36 Months**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
MLP	21.12	15.14	62.4	45.47	107.9	44.71
PLS	23.1	20.51	48.55	17.19	97.83	37.89
MLP-PLS	18.09	14.08	64.43	44.76	105.5	43.09
MLP-AE ($n = 2$)	8.4	7.64	7.89	3.77	21.52	17.39
MLP-AE ($n = 3$)	4.63	2.12	5.79	1.73	10.04	6.14
MLP-AE ($n = 4$)	7.77	8.76	7.84	7.33	10.22	8.27
MLP-CFE ($n = 2$)	8.26	4.66	8.6	4.22	15.17	8.35
MLP-CFE ($n = 3$)	3.73	1.65	7.12	3.11	9.05	3.62
MLP-CFE ($n = 4$)	9.51	8.23	9.85	7.44	14.69	8.6
MLP-tuned	21.12	15.14	51.759	54.83	21.99	22.03

The training times per epoch, averaged for six gases, were compared for each model and are shown in Table 4. The results for the MLP-CFE and MLP-AE models with latent vector dimensions of $n = 2, 3, 4$ were averaged to represent the time complexity of MLP-CFE and MLP-AE. MLP-PLS has the longest training time per epoch, at an average of 0.239 s, followed by MLP and MLP-tuned at 0.1592 s and 0.132 s, respectively. The average training time per epoch is 0.0912 s and 0.0084 s for MLP-CFE and MLP-AE, respectively, both below 0.1 s, which is lower than training times for MLP, MLP-PLS, and MLP-tuned. These results

indicate that, despite the presence of a decoder part during training of the encoder module, our proposed method outperforms other methods, including the fine-tuning method, in terms of time complexity.

Table 4. The training time per epoch (s/epoch) for each model.

	MLP-CFE	MLP-AE	MLP	MLP-PLS	MLP-Tuned
time (s/epoch)	0.0912	0.084	0.1592	0.239	0.132

4. Conclusions

In this paper, we introduce an effective method for compensating for sensor drift using deep learning algorithms. Our method involves concatenating calibration feature vectors, embedded from a trained calibration feature encoder as prompts, with sensor data, which are then used as inputs to the neural network during training. The CFE receives transfer samples as inputs and encodes the calibration feature vectors which contain sensor drift information, allowing the neural network to learn sensor drifts and output a drift-compensated label based on a new sensor signal. We evaluate the effectiveness of this method by testing it on a 3-year dataset of gas sensor array drifts for six types of gases at different concentrations. Our results show that our method achieves a higher accuracy than a conventional MLP, an MLP with other feature extractors, and even an MLP with additional fine tuning. Our work verifies that representation learning methods like masked autoencoders can efficiently represent sensor drift information from transfer samples that have a complex and nonlinear structure. In addition, we show that training a neural network with informative prompts is effective in terms of model performance. As far as the authors are aware, the proposed methodology is the first proposal of compensating for sensor drift in deep learning models without-fine tuning. The research community can develop this prompt-based learning approach in future studies, leading to active research in the area. In this field, classifying gas types is just as important as gas concentration estimations; however, in this research, this method was only validated via concentration estimations, constituting a limitation. Furthermore, this methodology has only been validated on regression models based on multi-layer perceptrons. Thus, future work will focus on extending the proposed methodology across other tasks and various types of deep learning models.

Author Contributions: Conceptualization, S.K., J.-H.P., H.-D.J. and H.N.; methodology, S.K.; software, S.K.; validation, S.K. and H.-D.J.; formal analysis, S.K., J.-H.P., H.-D.J. and H.N.; investigation, S.K. and J.-H.P.; resources, S.K., J.-H.P. and H.N.; data curation, S.K., J.-H.P., H.-D.J. and H.N.; writing—original draft preparation, S.K., J.-H.P., H.-D.J. and H.N.; writing—review and editing, D.E.C.; visualization, S.K.; supervision, D.E.C.; project administration, D.E.C.; funding acquisition, D.E.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Agency For Defense Development Grant funded by the Korean Government (UI220039ZD).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: A publicly available dataset was analyzed in this study. The dataset can be found at: https://doi.org/10.24432/C5RP6W (19 March 2024).

Conflicts of Interest: Author Jae-Hyeon Park was employed by the company Samsung Display Co. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CFE	Calibration Feature Encoder
MLP	Multi Layer Perceptron
PLS	Partial Least Square
RMSE	Root Mean Squared Error
SD	Standard Deviation

References

1. Pareek, V.; Chaudhury, S.; Singh, S. Handling non-stationarity in E-nose design: A review. *Sens. Rev.* **2021**, *42*, 39–61.
2. Karakaya, D.; Ulucan, O.; Turkan, M. Electronic nose and its applications: A survey. *Int. J. Autom. Comput.* **2020**, *17*, 179–209.
3. Połap, D.; Srivastava, G.; Jaszcz, A. Energy consumption prediction model for smart homes via decentralized federated learning with LSTM. *IEEE Trans. Consum. Electron.* **2023**, 1. https://doi.org/10.1109/TCE.2023.3325941.
4. Haugen, J.E.; Tomic, O.; Kvaal, K. A calibration method for handling the temporal drift of solid state gas-sensors. *Anal. Chim. Acta* **2000**, *407*, 23–39.
5. Rudnitskaya, A. Calibration update and drift correction for electronic noses and tongues. *Front. Chem.* **2018**, *6*, 433.
6. Laref, R.; Losson, E.; Sava, A.; Adjallah, K.; Siadat, M. A comparison between SVM and PLS for E-nose based gas concentration monitoring. In Proceedings of the 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 19–22 February 2018; IEEE: New York, NY, USA, 2018; pp. 1335–1339.
7. Ye, Z.; Liu, Y.; Li, Q. Recent progress in smart electronic nose technologies enabled with machine learning methods. *Sensors* **2021**, *21*, 7620.
8. Jang, H.D.; Park, J.H.; Nam, H.; Chang, D.E. Deep neural networks for gas concentration estimation and the effect of hyper-parameter optimization on the estimation performance. In Proceedings of the 2022 22nd International Conference on Control, Automation and Systems (ICCAS), Busan, Republic Korea, 27 November–1 December 2022; IEEE: New York, NY, USA, 2022; pp. 15–19.
9. Caterini, A.L.; Chang, D.E. *Deep Neural Networks in a Mathematical Framework*; Springer: Berlin/Heidelberg, Germany, 2018.
10. Wang, X.; Zhao, W.; Ma, R.; Zhuo, J.; Zeng, Y.; Wu, P.; Chu, J. A novel high accuracy fast gas detection algorithm based on multi-task learning. *Measurement* **2024**, *228*, 114383.
11. Zhang, L.; Liu, Y.; He, Z.; Liu, J.; Deng, P.; Zhou, X. Anti-drift in E-nose: A subspace projection approach with drift reduction. *Sens. Actuators Chem.* **2017**, *253*, 407–417.
12. Cheng, Y.C.; Chou, T.I.; Chiu, S.W.; Tang, K.T. A concentration-based drift calibration transfer learning method for gas sensor array data. *IEEE Sens. Lett.* **2020**, *4*, 7003704.
13. Pareek, V.; Chaudhury, S.; Singh, S. Hybrid 3DCNN-RBM network for gas mixture concentration estimation with sensor array. *IEEE Sens. J.* **2021**, *21*, 24263–24273.
14. Pareek, V.; Chaudhury, S. Deep learning-based gas identification and quantification with auto-tuning of hyper-parameters. *Soft Comput.* **2021**, *25*, 14155–14170.
15. Feng, L.; Dai, H.; Song, X.; Liu, J.; Mei, X. Gas identification with drift counteraction for electronic noses using augmented convolutional neural network. *Sens. Actuators Chem.* **2022**, *351*, 130986.
16. Ma, Z.; Luo, G.; Qin, K.; Wang, N.; Niu, W. Online sensor drift compensation for E-nose systems using domain adaptation and extreme learning machine. *Sensors* **2018**, *18*, 742.
17. Liu, T.; Li, D.; Chen, Y.; Wu, M.; Yang, T.; Cao, J. Online drift compensation by adaptive active learning on mixed kernel for electronic noses. *Sens. Actuators Chem.* **2020**, *316*, 128065.
18. Se, H.; Song, K.; Sun, C.; Jiang, J.; Liu, H.; Wang, B.; Wang, X.; Zhang, W.; Liu, J. Online drift compensation framework based on active learning for gas classification and concentration prediction. *Sens. Actuators Chem.* **2024**, *398*, 134716.
19. Petroni, F.; Rocktäschel, T.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A.H.; Riedel, S. Language models as knowledge bases? *arXiv* **2019**, arXiv:1909.01066.
20. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
21. Schick, T.; Schütze, H. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv* **2020**, arXiv:2001.07676.
22. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
23. Shin, T.; Razeghi, Y.; Logan IV, R.L.; Wallace, E.; Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv* **2020**, arXiv:2010.15980.
24. Gao, T.; Fisch, A.; Chen, D. Making pre-trained language models better few-shot learners. *arXiv* **2020**, arXiv:2012.15723.
25. Liu, X.; Ji, K.; Fu, Y.; Tam, W.; Du, Z.; Yang, Z.; Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Dublin, Ireland, 22–27 May 2022; pp. 61–68.

26. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *AI Open* **2023**. *in press*, https://doi.org/10.1016/j.aiopen.2023.08.012
27. Vergara, A.; Vembu, S.; Ayhan, T.; Ryan, M.A.; Homer, M.L.; Huerta, R. Chemical gas sensor drift compensation using classifier ensembles. *Sens. Actuators Chem.* **2012**, *166*, 320–329.
28. Lu, S.; Guo, J.; Liu, S.; Yang, B.; Liu, M.; Yin, L.; Zheng, W. An improved algorithm of drift compensation for olfactory sensors. *Appl. Sci.* **2022**, *12*, 9529.
29. Zhang, L.; Zhang, D. Domain adaptation extreme learning machines for drift compensation in E-nose systems. *IEEE Trans. Instrum. Meas.* **2014**, *64*, 1790–1801.
30. Jiang, Q.; Zhang, Y.; Zhang, Y.; Liu, J.; Xu, M.; Ma, C.; Jia, P. An Adversarial Network Used for Drift Correction in Electronic Nose. Available online: https://ssrn.com/abstract=4725735 (19 March 2024)
31. Clifford, P.; Tuma, D. Characteristics of semiconductor gas sensors I. Steady state gas response. *Sens. Actuators* **1982**, *3*, 233–254.
32. Sinha, S.; Bhardwaj, R.; Sahu, N.; Ahuja, H.; Sharma, R.; Mukhiya, R. Temperature and temporal drift compensation for Al2O3-gate ISFET-based pH sensor using machine learning techniques. *Microelectron. J.* **2020**, *97*, 104710.
33. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
34. Zhang, S.; Wang, B.; Li, X.; Chen, H. Research and application of improved gas concentration prediction model based on grey theory and BP neural network in digital mine. *Procedia CIRP* **2016**, *56*, 471–475.
35. AlOmar, M.K.; Hameed, M.M.; AlSaadi, M.A. Multi hours ahead prediction of surface ozone gas concentration: robust artificial intelligence approach. *Atmos. Pollut. Res.* **2020**, *11*, 1572–1587.
36. Hulland, J. Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strateg. Manag. J.* **1999**, *20*, 195–204.
37. Wold, S.; Trygg, J.; Berglund, A.; Antti, H. Some recent developments in PLS modeling. *Chemom. Intell. Lab. Syst.* **2001**, *58*, 131–150.
38. Mateos-Aparicio, G. Partial least squares (PLS) methods: Origins, evolution, and application to social sciences. *Commun. Stat.-Theory Methods* **2011**, *40*, 2305–2317.