Correcting Instrumental Variation and Time-varying Drift: A Transfer Learning Approach with Autoencoders

Ke Yan and David Zhang, *Fellow, IEEE*

***Abstract*—Electronic noses (e-noses) are instruments that can be used to measure gas samples conveniently. Based on the measured signal, the type and concentration of the gas can be predicted by pattern recognition algorithms. However, e-noses are often affected by influential factors such as instrumental variation and time-varying drift. From the viewpoint of pattern recognition, the factors make the posterior distribution of the test data drift from that of the training data, thus will degrade the accuracy of the prediction models. In this paper, we propose drift correction autoencoder (DCAE) to address this problem. DCAE learns to model and correct the influential factors explicitly with the help of transfer samples. It generates drift-corrected and discriminative representation of the original data, which can then be applied to various prediction algorithms. We evaluate DCAE on datasets with instrumental variation and complex time-varying drift. Prediction models are trained on samples collected with one device or in the initial time period, then tested on other devices or time periods. Experimental results show that DCAE outperforms typical drift correction algorithms and autoencoder-based transfer learning methods. It can improve the robustness of e-nose systems and greatly enhance their performance in real-world applications.**

***Index Terms*—Autoencoder, calibration transfer, electronic nose, drift correction, spectroscopy, transfer learning**

I. Introduction

ODOR sensing with electronic noses (e-noses), which is known as machine olfaction, has received much interest from researchers. Based on the signals measured with e-noses, the type and concentration of gases can be predicted by pattern recognition algorithms [1]. The applications of e-noses range from agriculture and food to environmental monitoring, robotics, biometrics, and disease analysis [2], [3], [4], [5]. Although e-noses are convenient to use, fast, and cheap, their robustness still remains a problem [6], [7]. The chemical sensor array is the key part of an e-nose. However, most chemical sensors are influenced by some factors which can make their signals irreproducible. For example, because of the variations in the fabrication of sensors and devices, the responses to the same signal source may not be identical for different e-noses. In addition, the sensitivity characteristics of sensors, the operating condition, or even the signal source itself, can change over time. Because of the instrumental variation and the time-varying drift, the accuracy of initially trained prediction models degrades in the test stage. This issue also exists in many other devices and sensor systems, such as in spectrometers [8]. It hampers the usage of the sensor systems in real-world applications.

Following the terms in transfer learning [9], we assume that the training samples are drawn from a source domain (e.g. collected with the old device or in the initial time period), whereas the test samples are drawn from target domains (e.g. collected with new devices or in later time periods). The influential factors mentioned above cause the posterior data distribution in target domains to drift from that in the source domain. Many algorithms have been proposed to correct the drift and improve the robustness of the prediction models. It is worth noting that in the field of spectroscopy, devices generate similar one-dimensional signals and suffer the similar problems of robustness like e-noses, so the algorithms in both areas often work universally.

An intuitive and commonly-used idea is to transform the features in target domains to match those in the source one, so that the transformed target samples can be predicted by the models trained in the source domain [4], [10], [8], [11], [12], [13]. Algorithms in this category often seek help from transfer samples, which are a small set of predefined gas samples collected from every domain separately. The correspondence relationship between transfer sample pairs provides useful mapping information between two domains. Efforts have also been made in the model-level for drift correction. Prediction models suitable for the target samples were learned based on labeled source samples and a few labeled [14], [15], [16] or unlabeled [17] target samples or transfer samples [18]. These methods showed better accuracy than the feature-level ones.

Algorithms mentioned above operate on traditional hand-crafted features. In recent years, unsupervised feature/representation learning methods based on deep networks have achieved promising results [19], [20], [21], [22]. These methods make use of the plentiful unlabeled data to learn representative features, whose discriminative power can be further enhanced by supervised fine-tuning. The adoption of nonlinear activation functions and multi-layer stacking strategy enables the learned features to capture complex structures in the data. In the field of machine

The work is partially supported by the GRF fund from the HKSAR Government, the central fund from Hong Kong Polytechnic University, the NSFC fund (61332011, 61272292, 61271344), Shenzhen Fundamental Research fund (JCYJ20150403161923528, JCYJ20140508160910917), and Key Laboratory of Network Oriented Intelligent Computation, Shenzhen, China.

K. Yan is with the Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: yank10@mails.tsinghua.edu.cn).

D. Zhang is with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Department of Computing, Biometrics Research Centre, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk).

olfaction, the pioneer works by Längkvist et al. [23], [24] have shown that features learned by an autoencoder or restricted Boltzmann machines (RBMs) outperform traditional e-nose features. Transfer learning with deep networks has been discussed in [20], [25], [26], [27], [28], [29], [30]. However, there is still no literature working on explicitly correcting instrumental variation and time-varying drift in sensor systems with deep networks. Furthermore, most transfer learning methods are designed for discrete source and target domains, whereas in the problem of time-varying drift, it is not easy to split data into such domains because the drift is continuous in time.

In this paper, we propose drift correction autoencoder (DCAE) for joint representation learning and drift correction. Besides the original features, the "domain features" are also inputted into DCAE, which contain the information about when and with which device the sample was collected. They make it convenient for DCAE to handle both the discrete drift among devices and the continuous drift along with time. DCAE explicitly models the influence of these factors to the learned representation with the aid of transfer samples. A correction layer is used to further enhance DCAE's ability to correct complex time-varying drift. The hidden representation of DCAE is drift-corrected and can then be applied to various prediction algorithms. The supervised term in its objective function makes the representation to be discriminative as well.

The rest of the paper is organized as follows. Related work on drift correction and autoencoders is briefly reviewed in Section II. Section III describes the proposed DCAE in detail. Section IV sketches the transfer sample selection algorithm used in this paper. The experimental configurations and results are presented in Section V, along with some discussions. Section VI concludes the paper.

II. Related Work

In this section, we will first introduce typical drift correction algorithms in machine olfaction and spectroscopy. Note that the term "drift" often refers to time-varying drift caused by sensor aging, sensor poisoning, etc. [6]. In this paper, we use the term to indicate the change in data distribution, which can be caused by instrumental variation, sensor aging, environmental change, and so on. Algorithms based on variable standardization are simple and widely-used [4], [10], [8]. Regression models are built on the transfer samples to transform the features in target domains to match those in the source one. Then, the transformed target samples can be directly predicted by the models trained in the source domain. Algorithms based on component correction (CC) [11], [12] correct drift by removing undesired components from data. CC-PCA [11] applies principal component analysis to the transfer samples to find the component of drift. Orthogonal signal correction (OSC) [12] pools samples with and without drift, then finds the undesired components by estimating the subspace that is orthogonal to the labels. The drawback of CC-like methods is that when the drift is complex, it is not easy to well separate the component of drift from that of the useful information [13], [18].

Motivated by the progress in transfer learning and concept drift adaptation, many algorithms have emerged in recent years. Vergara et al. [14] adopted an ensemble strategy to cope with time-varying drift in e-noses. Multiple prediction models trained on samples in earlier batches are used to predict the sample collected in a later batch. Then, the predictions are fused by weighted majority voting. This algorithm as well as many concept drift adaptation algorithms [31] relies on newly arrived labeled data to update the prediction models, but the collection of labels is often a demanding job in real-world applications. Yu et al. [16] applied regularized multitask learning (RMTL) to transfer models between different spectrometers. Models of different devices are learned jointly with a prior belief that the models should resemble each other. This strategy allows knowledge transfer from the source to target domains, which is beneficial because labeled samples are often insufficient in target domains. Zhang et al. [15] used the source and the target data together to train an extreme learning machine. The method achieved promising accuracy in the dataset in [14] with the help of 30 labeled target samples. Transfer-sample-based coupled task learning (TCTL) was proposed in [18]. It uses a similar MTL strategy as in [16], but the knowledge transfer is based on transfer samples instead of labeled samples. Moreover, a regularization term is used to increase the weight of more robust features. These model-level methods generally have better accuracy than the feature-level ones. Transductive transfer learning and semi-supervised learning have also been adopted for drift correction, such as the methods based on geodesic flow kernels and manifold regularization [17]. They have the advantage of only requiring unlabeled target samples, but their accuracy is still not satisfactory.

Autoencoders have been used to learn features from e-nose signals [23], [24]. When the data contains drift, transfer learning with autoencoders can be considered [20], [25], [26], [27], [28], [29], [30]. Kandaswamy et al. [29] proposed to reuse part of the weights of the network trained with source data and fine-tune the other part with labeled target samples. There are many other works that do not need supervised information in target domains. [25] and [20] suggested that the learned features could generalize in multiple domains if one train an autoencoder with unlabeled data from all these domains. Chopra et al. [30] presented an interesting idea to learn an universal representation across domains. They constructed intermediate domains between the source and the target ones by merging different proportion of samples from the two domains. The final representation was the concatenation of the features learned from each domain, which contained the information of the distributional shift across domains. In the work by Deng et al. [28], the knowledge in target domains is transferred to the source one by a similarity constraint of the weights.

Although these three methods can make the learned features capture the information in multiple domains, they may encounter difficulties in the scenario of drift correction, because the difference in distribution across domains is not explicitly reduced. The strategies used in [26], [27], [32] tried to reduce the difference. Zhou et al. [26] first learned features from

both domains separately with autoencoders, then computed a transformation matrix between the learned features according to a set of "cross-domain parallel data". This method is similar to variable standardization in machine olfaction. In [27], the mean of the learned feature vectors of both domains were required to have small KL divergence. Kan et al. [32] designed a network to learn pose-corrected features for face recognition. In the training process, face images with pose variations were used to reconstruct those without pose variations. The training images can actually be viewed as transfer samples.

III. Drift correction autoencoder (DCAE)

A. Preliminary: autoencoder

In this paper, we will use the letter m to denote the number of features, h for the number of hidden units in the network, and n for the number of samples and so on. The basic framework of an autoencoder is essentially a feed-forward neural network with one hidden layer. It takes an input vector $\boldsymbol{x} \in \mathbf{R}^m$, encodes it to a new representation $\boldsymbol{z} \in \mathbf{R}^h$, and then decodes $\boldsymbol{z}$ to $\hat{\boldsymbol{x}} \in \mathbf{R}^m$ in the original space. The encoding and decoding process can be formulated as:

$$\boldsymbol{z} = \sigma(W\boldsymbol{x} + \boldsymbol{b}), \tag{1}$$

$$\hat{\boldsymbol{x}} = \sigma(W'\boldsymbol{z} + \boldsymbol{b}'), \tag{2}$$

where $W \in \mathbf{R}^{h\times m}$ and $W' \in \mathbf{R}^{m\times h}$ are the weight matrices, $\boldsymbol{b} \in \mathbf{R}^h$ and $\boldsymbol{b}' \in \mathbf{R}^m$ are the bias vectors. σ is an activation function such as sigmoid, hyperbolic tangent (tanh), or linear (i.e. using an identity function). The objective of an autoencoder is to minimize the reconstruction error [21]:

$$J_{AE}(W, \boldsymbol{b}, W', \boldsymbol{b}') = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2, \tag{3}$$

where n is the number of training samples. When the number of hidden units h is less than the dimension of the original feature vector m, there will be a "bottleneck" in the network and a compressed representation of the original features will be learned. Note that one can use a tied weight [21] by defining $W' = W^{\mathrm{T}}$, which can reduce the number of parameters to estimate.

One variant of the basic autoencoder is the denoising autoencoder [19]. It first corrupts the input $\boldsymbol{x}$ into $\tilde{\boldsymbol{x}}$ by randomly setting some of the features to zero. After obtaining the reconstruction of $\tilde{\boldsymbol{x}}$ from Eq. (1) and (2), one minimizes the difference between it and the clean $\boldsymbol{x}$. It is expected that under this strategy, the learned representation will be more stable and robust to corruptions of the input, as well as capture more useful structures in the input distribution [19]. The single-layer autoencoders can also be stacked into deeper ones to obtain more abstract representation [21]. The hidden representation of an outer autoencoder serves as the input to an inner one. The hidden representation of the innermost one is regarded as the final learned representation, which can then be applied to various prediction algorithms.

B. Basic strategies

Most transfer learning algorithms [9] split data into one or more source domains and one or more target ones. Labeled samples are sufficient in source domains, but scarce or not available in target ones. In drift correction problems, data without and with drift are often regarded as source and target domains, respectively [15]. For example, when the data are collected with different devices, each device defines a domain. In the case of time-varying drift, one can split the data into several batches in chronological order and treat each batch as a domain. Within each domain, drift is small. Prediction models need to be transferred from the source domain (the initial device or batch) to the target domains (the subsequent devices or batches).

However, in the most general cases, data are collected in continuous streams. The amount of drift in each sample is different. This information of continuity will be lost if we force the data into batches. Therefore, instead of putting data into different domains according to which device it is from and when it was collected (acquisition time), we consider all data as a whole and design "domain features" for each sample to describe this information conveniently. If we only consider the instrumental variation, an one-hot coding scheme can be used. Suppose there are n_{dev} devices. The domain feature vector is thus $\boldsymbol{d} \in \mathbf{R}^{n_{dev}}$, where $d_i = 1$ if the sample is from the ith device and 0 otherwise. If the time-varying drift is also considered, the acquisition time can be further added into $\boldsymbol{d}$. Suppose a gas sample is collected from the ith device at time t, then $\boldsymbol{d} \in \mathbf{R}^{2n_{dev}}$ and

$$d_j = \begin{cases} 1, & j = 2i-1, \\ t, & j = 2i, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Transfer samples are frequently used in the field of machine olfaction and spectroscopy [4], [10], [8], [11], [13], [18]. They are often a group of standard gases with predefined types and concentrations that are reproducible and easy to acquire. Existing transfer learning methods mostly rely on labeled or unlabeled samples from target domains for knowledge transfer. Compared to unlabeled target samples, transfer samples are more informative, because they contain the mapping information between domains. Meanwhile, they are more convenient to obtain than labeled target samples in many real-world applications. Note that transfer samples are unlabeled. They can be either of the same type with the samples to predict or not.

In order to correct instrumental variation and time-varying drift, transfer samples should be collected from each device periodically. They are viewed as representatives or milestones of the samples collected with the same device and in the similar time period. It is natural to assign the first group of transfer samples collected by the first device as the reference group (source). All other groups collected with other devices or in later time periods (target) should be aligned with the reference. If the difference of representation is small between the reference and the other groups, one can expect that the drift has been reduced in the learned representation.

C. Formulation

Four sets of data are used in the learning procedure of DCAE. $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ contains both labeled and unlabeled data. The labeled data and their labels are included in $\mathcal{D}_L = \{(\boldsymbol{x}_j^{(L)}, \boldsymbol{y}_j^{(L)})\}_{j=1}^{n_L}$. n and n_L are the total number of samples and the number of labeled samples, respectively. $\mathcal{D}_S = \{\boldsymbol{t}_p^{(S)}\}_{p=1}^{n_{t,total}}$ and $\mathcal{D}_T = \{\boldsymbol{t}_q^{(T)}\}_{q=1}^{n_{t,total}}$ denote the source and target transfer samples, respectively. $n_{t,total}$ is the total number of transfer sample pairs. $\boldsymbol{x}, \boldsymbol{x}^{(L)}, \boldsymbol{t}^{(S)}, \boldsymbol{t}^{(T)} \in \mathbf{R}^m$. $\boldsymbol{y}^{(L)} \in \mathbf{R}^c$, where c is the number of labels. Each sample $(\boldsymbol{x}_i, \boldsymbol{x}_j^{(L)}, \boldsymbol{t}_p^{(S)}, \boldsymbol{t}_q^{(T)})$ has a corresponding domain feature vector $(\boldsymbol{d}_i, \boldsymbol{d}_j^{(L)}, \boldsymbol{d}_p^{(S)}, \boldsymbol{d}_q^{(T)})$.

To clarify the composition of $\mathcal{D}_S$ and $\mathcal{D}_T$, let us assume that each group of transfer samples is made up of $n_{t,gas}$ different gas samples, and $n_{t,group}$ groups were collected altogether. Thus, $\mathcal{D}_T$ is the stack of samples in all groups, whereas $\mathcal{D}_S$ is the stack of $n_{t,group}$ repetitions of the samples in the reference group. Therefore, each target transfer sample $\boldsymbol{t}_p^{(T)}$ has a corresponding source one $\boldsymbol{t}_p^{(S)}$. The total number of transfer sample pairs is $n_{t,total} = n_{t,gas} n_{t,group}$. DCAE can thus incorporate the information from all available transfer sample groups into the learning process.

On the basis of an autoencoder, we utilize a new weight matrix to correct drift. It is denoted as $W_D \in \mathbf{R}^{h \times m_d}$, where m_d is the length of the domain feature vector and h is the number of hidden units of the original autoencoder. The component of drift is explicitly removed from the hidden representation by adding $W_D\boldsymbol{d}$ in the encoding process, and recovered in the decoding process by subtracting it:

$$\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{d}) = \sigma(W\boldsymbol{x} + W_D\boldsymbol{d} + \boldsymbol{b}), \quad (5)$$

$$\hat{\boldsymbol{x}} = g(\boldsymbol{z}, \boldsymbol{d}) = \sigma(W'(\boldsymbol{z} - W_D\boldsymbol{d}) + \boldsymbol{b}'). \quad (6)$$

Through the equations above, drift correction in the hidden representation and data reconstruction can both be accomplished. If a linear activation function is used, the reconstruction will be identical to that in the original autoencoder.

The objective function of DCAE is expressed as follows:

$$\begin{aligned} J_{DCAE}(W, \boldsymbol{b}, W', \boldsymbol{b}', W_D, W_S, \boldsymbol{b}_S) = \\ \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - g(f(\boldsymbol{x}_i, \boldsymbol{d}_i), \boldsymbol{d}_i)\|^2 \\ + \frac{\lambda_1}{n_L}\sum_{j=1}^{n_L} \mathcal{L}\left(f(\boldsymbol{x}_j^{(L)}, \boldsymbol{d}_j^{(L)}), \boldsymbol{y}_j^{(L)}; W_S, \boldsymbol{b}_S\right) \\ + \frac{\lambda_2}{n_{t,total}}\sum_{p=1}^{n_{t,total}} \|f(\boldsymbol{t}_p^{(S)}, \boldsymbol{d}_p^{(S)}) - f(\boldsymbol{t}_p^{(T)}, \boldsymbol{d}_p^{(T)})\|^2 \end{aligned} \quad (7)$$

It can be decomposed into three parts: the first term is the reconstruction error of all data; the second term is the supervised loss of the labeled samples; the third term is the alignment error of the transfer samples. λ_1, λ_2 and μ are regularization hyper-parameters. The weight matrix $W_S \in \mathbf{R}^{c \times h}$ and bias vector $\boldsymbol{b}_S \in \mathbf{R}^c$ are used in the supervised model. By minimizing Eq. (7), all weight matrices and bias vectors are learned simultaneously.

The first term of Eq. (7) is similar to the objective of the original autoencoder, except that the functions f and g are those defined in Eq. (5) and (6), which take drift into consideration. The second term incorporates the label information into the function. The loss function $\mathcal{L}$ can be determined according to the problem. For example, the softmax function [27] can be used for classification problems, whereas the squared error function can be used for regression ones.

The third term is key for drift reduction in DCAE. It requires the corresponding source and target transfer samples to have similar hidden representations. The correction weight matrix W_D will be learned mainly base on this term. W and $\boldsymbol{b}$ will also be influenced to extract features that are more robust to drift. An alternative strategy to learn W_D is using the drift-corrected hidden representation of the target transfer samples to reconstruct the corresponding source ones, which is similar to the strategy in [32]. The accuracy of this strategy is not as good as that of the proposed one, which is probably because an trained autoencoder only has low reconstruct error on certain data [21], which are those in $\mathcal{D}$. However, transfer samples are not always of the same type with data in $\mathcal{D}$. For example, one is standard gas and the other is breath sample. Therefore, the transfer samples may not be well reconstructed.

D. Handling complex time-varying drift

For a sample with domain feature $\boldsymbol{d}$, the amount of correction received by the ith hidden unit is the ith element of $W_D\boldsymbol{d}$. This value is device-specific and linearly proportional to the acquisition time (see the coding scheme of $\boldsymbol{d}$ in Section III-B). However, the relationship between time and the time-varying drift in most real-world applications is nonlinear. The sensitivity characteristic of gas sensors can be affected by many factors such as aging effect, humidity, temperature, background change, and sensor replacement [6]. Thus, the drift can display a complex pattern, which makes the linear correction insufficient. To enhance the correction ability, we further insert a correction layer between the domain feature vector and the hidden layer in the basic DCAE described in the last section. The encoding and decoding process thus becomes:

$$\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{d}) = \sigma(W\boldsymbol{x} + W_{D1}\,\sigma_{cor}(W_{D0}\boldsymbol{d}) + \boldsymbol{b}), \quad (8)$$

$$\hat{\boldsymbol{x}} = g(\boldsymbol{z}, \boldsymbol{d}) = \sigma(W'(\boldsymbol{z} - W_{D1}\,\sigma_{cor}(W_{D0}\boldsymbol{d})) + \boldsymbol{b}'). \quad (9)$$

In the equation above, $W_{D0} \in \mathbf{R}^{h_{cor} \times m_d}$ is the weight matrix from the domain features to the correction layer, where the number of units in the correction layer is h_{cor}. $W_{D1} \in \mathbf{R}^{h \times h_{cor}}$ is the weight matrix from the correction layer to the hidden layer of DCAE. σ_{cor} is the activation function of the correction layer. It should be set to a nonlinear one, as a linear one makes the correction linearly related to time, which is equivalent to the basic DCAE. Ideally, the correction output $W_{D1}\,\sigma_{cor}(W_{D0}\boldsymbol{d})$ should compensate the drift of each hidden unit with regard to time and device. There is no bias vector in the correction layer. The 1's in the domain features in Eq. (4) can be viewed as device-specific constant terms, so the bias vector is merged into W_{D0}. The number of units in the correction layer h_{cor} should be selected according to the complexity of the time-varying drift. A larger h_{cor} is preferable for more complex drift.

Both the basic DCAE and that with correction layer explicitly models the influence of instrumental variation and time-varying drift by the weight matrix W_D (or W_{D0} and W_{D1}). The parameters for different devices are stored in different columns of W_D or W_{D0}. Similar to the stacked autoencoders, the number of hidden layers in DCAE can be increased to extract more abstract representation. The correction output is added before the first hidden layer and removed after the last hidden layer, so that the representations learned in the hidden layers are all drift-corrected.

E. Summary

Fig. 1 illustrates the architecture of a DCAE with correction layer and three hidden layers. The solid arrows indicate the flow of signals. The original features are input to the first layer of the main body of the network (in blue), whereas the domain features are input to the first layer of the correction part (in orange). The output of the correction layer is added to the input of the second layer of the main body, in the meantime subtracted from the output of the fourth (the second to the last) layer. The hidden representation is applied to predict for the labeled samples and align the transfer samples, see the objective function Eq. (7).

It is important to carefully initialize the weights in deep networks. The greedy layerwise unsupervised pretraining strategy [21] is widely-used for stacked autoencoders. We first pretrain a stacked denoising autoencoder with all samples in $\mathcal{D}$. The fraction of corrupted input is set to 10% in this paper. The weight tying strategy is used. Then, supervised fine-tuning [19] is performed with the labeled data in $\mathcal{D}_L$. This step minimizes the supervised loss of the labeled data. Although a supervised loss term has been included in the objective function of DCAE, we find that this fine-tuning step can further improve the accuracy because it can make the initialized weights in DCAE closer to the optimal solution. An experiment will be made in Section V-D to compare the impact of different training procedures to the accuracy. Finally, the weights and biases in the fine-tuned network are used to initialize $W, \boldsymbol{b}, W', \boldsymbol{b}'$ in DCAE. W_S, W_{D1} are randomly initialized. $\boldsymbol{b}'_S, W_{D0}$ are initialized to zeros. After that, Eq. (7) can be optimized using methods such as gradient descent or conjugate gradient. Note that the denoising strategy is not used in DCAE.

Once the optimization is done, the labeled samples and their domain features can be input to DCAE. Their hidden representations and labels are used to train a separate prediction model, which is then applied on the hidden representations of the unlabeled samples to obtain the predicted labels. The entire process is summarized in Algorithm III.1.

Algorithm III.1 Drift correction autoencoder (DCAE)

Input: The unlabeled dataset $\mathcal{D}$, the labeled dataset $\mathcal{D}_L$, and the transfer sample datasets $\mathcal{D}_S, \mathcal{D}_T$. The device index and acquisition time of each sample are known.

Output: The predicted labels of samples in $\mathcal{D}$.

1: Pretrain a stacked denoising autoencoder unsupervisedly with $\mathcal{D}$, and then fine-tune it with $\mathcal{D}_L$;
2: Initialize the weights in DCAE based on the weights of the autoencoder;
3: Create the domain feature vector for each sample according to Eq. (4);
4: Optimize Eq. (7) with samples in $\mathcal{D}, \mathcal{D}_L, \mathcal{D}_S, \mathcal{D}_T$ and their domain features to obtain the weights of DCAE;
5: Train a prediction model with the hidden representations of $\mathcal{D}_L$, and then apply it on the hidden representations of $\mathcal{D}$.

IV. Selection of transfer samples

In transfer-sample-based methods, transfer samples should be representative in the dataset for effective knowledge transfer. Meanwhile, the number of transfer samples should be as small as possible to ease the burden of collecting them repeatedly. A common method is to collect a candidate set with the initial device at the training stage, then select a compact and representative group of transfer samples from the candidates [4], [8]. The selected samples can be measured by new devices and in later time periods. We utilize an unsupervised active learning method called locally linear reconstruction (LLR) [33] for transfer sample selection in this paper. In LLR, the most representative samples are those which can be used to best reconstruct the candidate set. By constraining that a data point can only be linearly reconstructed from its neighbors, LLR takes into account the local structure of the data space. The method has good performance in our experiments. Another related issue is to automatically choose the proper time to collect transfer samples. For this issue, one can refer to the change detection algorithms in [31]. The process of transfer sample collection of e-noses is discussed in [34].

V. Experiments

In this section, we conduct experiments on three datasets: one with time-varying drift, one with instrumental variation, and one with both. Comparison is made between DCAE and typical transfer learning methods on autoencoders and drift correction methods in the field of machine olfaction and spectroscopy. The proposed method was implemented based on the Theano library [35]. The optimization algorithm was conjugate gradient with the maximum iteration number set to 1000. No other strategies were used in the optimization process.

A. Gas sensor array drift dataset

The gas sensor array drift dataset was collected by Vergara et al. [14]. It is a public dataset[1] widely used for research on drift correction [14], [15], [17], [18]. An e-nose with 16 gas sensors was utilized to measure six kinds of gases (ammonia, acetaldehyde, acetone, ethylene, ethanol, and toluene) at different concentrations. A total of 13910 samples were collected over a course of 36 months. They are split into 10 batches in chronological order. The period of collection and the number

[1] http://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations

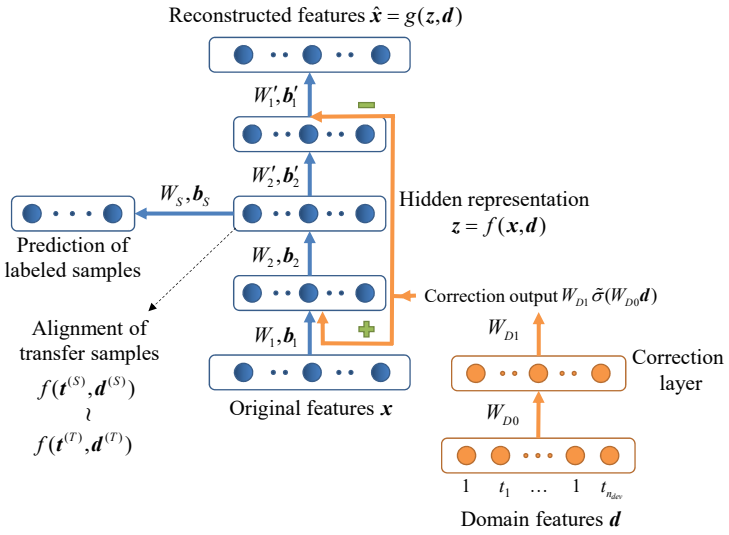

Fig. 1. Architecture of a DCAE with correction layer and three hidden layers.

TABLE I
PERIOD OF COLLECTION AND NUMBER OF SAMPLES IN THE GAS SENSOR ARRAY DRIFT DATASET [14].

Batch ID	Month	Ammonia	Acetaldehyde	Acetone	Ethylene	Ethanol	Toluene	#Total
1	1, 2	83	30	70	98	90	74	445
2	3, 4, 8-10	100	109	532	334	164	5	1244
3	11-13	216	240	275	490	365	0	1586
4	14,15	12	30	12	43	64	0	161
5	16	20	46	63	40	28	0	197
6	17-20	110	29	606	574	514	467	2300
7	21	360	744	630	662	649	568	3613
8	22, 23	40	33	143	30	30	18	294
9	24, 30	100	75	78	55	61	101	470
10	36	600	600	600	600	600	600	3600

of samples in each batch can be found in Table I. The goal is to classify the type of gases, despite their concentrations.

In order to evaluate the performance of the drift correction algorithms, we assume that the labels in batch 1 are known, whereas those in batches 2–10 are to be predicted. This evaluation strategy was also used in previous literatures and resembles the situation in real-world applications. The time-varying drift across batches can be visually inspected in Fig. 2(a). Samples in two classes are projected to a 2D subspace using PCA. It can be found that there is an obvious drift for samples in both classes as time elapses. Therefore, if the prediction model trained on batch 1 is applied to classify samples in later batches, the accuracy will degrade. It is worth noticing that the direction in which the two classes can be discriminated is close to the direction of drift (along the y-axis). In such cases, correction methods that seeking to find a drift-free latent space or trying to remove the component of drift will suffer a loss of useful information.

We developed a DCAE with three hidden layers (one hidden layer between the input layer and the hidden representation layer). The numbers of units were 30, 20, 30, respectively. Further increasing the number of layers or units cannot improve the accuracy. The tanh activation function was used. In the dataset, each sample is represented by 128 features extracted from the sensors' response curves [14]. The original features have unbounded values that cannot be well reconstructed by the bounded tanh function. Thus, before it was inputted into the network, each feature was normalized to have zero mean and unit variance, then divided by 2 to make most feature values range from -1 to 1. Directly mapping the minimum and maximum values to -1 and 1 was not considered because of the disturbance of outlier samples. The domain features were created according to Eq. (4). In this dataset, the number of devices $n_{dev} = 1$, and the acquisition time t is defined as the batch index minus one, e.g. $t = 5$ for a sample in batch 6.

The transfer samples are not directly provided in the dataset, hence need to be selected for each batch. The candidate set of batch k ($k = 2, \ldots, 10$) was defined as the overlapping samples in batch 1 and k, namely the samples of the same gas and concentration. Then, LLR was used to sequentially select $n_{t,gas}$ transfer samples from each candidate set. Following Algorithm III.1, the weights of a pretrained and fine-tuned denoising stacked autoencoder were used to initialize the DCAE. Then, for each batch k ($k = 2, \ldots, 10$), we used the labeled samples in batch 1 and the transfer sample groups from batch 1 to k to train the DCAE. After that, the hidden representations of batch 1 were adopted to train logistic regression classifiers with the one-vs-all strategy for multi-class classification. The hidden representations of the samples in batch k were predicted by the classifiers and an accuracy

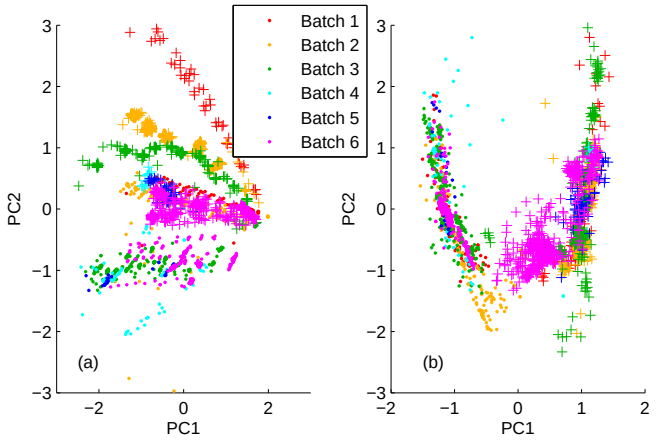

Fig. 2. Illustration of the drift across batches 1–6 in the gas sensor array drift dataset. Dots and plus signs represent ammonia and acetone samples, respectively. Different colors indicate different batches. The original features were used in plot (a) whereas the learned representation of DCAE were used in plot (b).

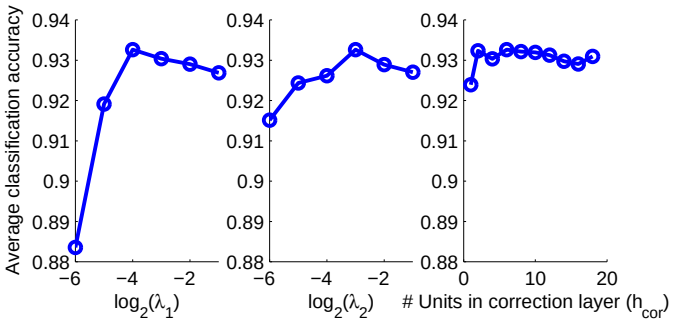

Fig. 3. Impact of the hyper-parameters on the average classification accuracy of DCAE-CL in the gas sensor array drift dataset.

was computed. Plot (b) of Fig. 2 displays the scatter of the hidden representations of the same group of samples as Fig. 2(a). We can find that the drift across batches has been greatly reduced and samples in each class are better clustered.

The proposed method has two variants: the basic DCAE (DCAE-basic) and DCAE with correction layer (DCAE-CL). In the latter variant, we set the activation function of the correction layer to be tanh as well. There are three major hyper-parameters in DCAE-CL, i.e. λ_1, λ_2, and the number of units in the correction layer h_{cor}. λ_1 controls the weight of the supervised loss term. λ_2 controls the weight of the transfer sample alignment error term. Larger h_{cor} brings higher capability in correcting complex time-varying drift. Their influence to the performance of DCAE-CL is investigated in Fig. 3. The number of transfer samples in each batch is $n_{t,gas} = 10$. When one hyper-parameter was tuned, the others were fixed. Owing to the random factor during the initialization of the networks, each experiment was repeated 10 times with 10 random seeds. The y-axis is the average accuracy of the 10 runs, where each accuracy is the average one on batches 2–10. It can be observed that when h_{cor} is not smaller than 2, the accuracy is relatively stable. Finally we adopt $\lambda_1 = 2^{-4}, \lambda_2 = 2^{-3}$ and $h_{cor} = 6$.

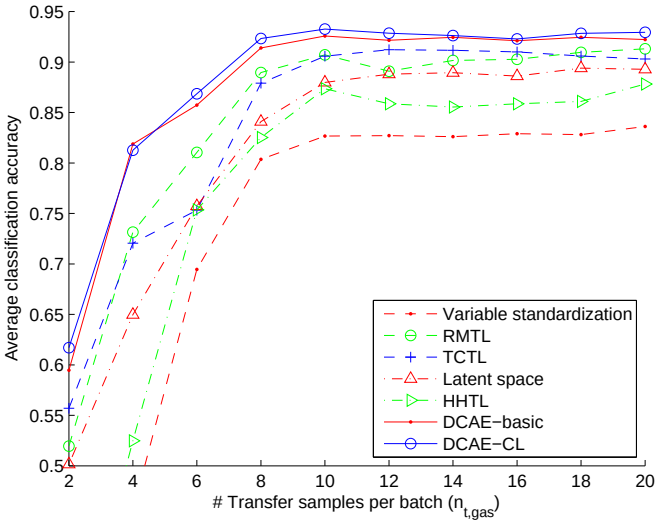

Fig. 4. Performance comparison on the gas sensor array drift dataset.

The proposed methods are compared with typical existing methods in Fig. 4 with regard to different numbers of transfer samples ($n_{t,gas}$). Variable standardization [18] transforms the original features in batches k ($k = 2, \ldots, 10$) to batch 1 using ridge regression based on transfer samples. Regularized multitask learning (RMTL) [16] jointly learns two models for batches 1 and k respectively based on the labeled samples in both batches. The transfer samples in batch k were used as labeled samples in this method. On the other hand, transfer-sample-based coupled task learning (TCTL) [18] is a multitask learning framework that seeks to align the transfer samples in the model level. Transfer learning with deep autoencoders (TLDA) [27] learns a common representation for the source and the target domains (batches 1 and k). It requires that the means of representations in the two domains are close. However, the method is suitable only when each domain has similar data composition. Moreover, transfer samples can provide much more information than the mean of each domain. Hence, we modify the method by replacing the mean of representations with transfer samples. We name the modified method as "latent space", because it actually assumes a latent representation space in which samples in different domains have similar distribution. The only difference between latent space and DCAE is that domain features are not used in the former method. Hybrid heterogeneous transfer learning (HHTL) [26] first uses marginalized stacked denoising autoencoders (mSDAs) to learn representations separately in each domain, then transform the target representation to the source one in the same way as in variable standardization. The representations learned in different layers are finally concatenated to an augmented feature vector. In all methods, the transfer samples were selected with LLR. The parameters of the autoencoders in latent space and HHTL were the same as those in DCAE. For the autoencoder-based methods, we report the average accuracy of 10 random runs.

From Fig. 4, we can find that the accuracy improves as $n_{t,gas}$ increases, but reaches a plateau when $n_{t,gas} = 10$.

The performance of variable standardization and HHTL is not promising, which is because the capacity of the feature-level correction methods is insufficient when facing complex drift. The fact that HHTL is better than variable standardization implies the augmented feature vector learned by mSDA is better than the original features. DCAE outperforms latent space, because it explicitly models and corrects drift, thus avoids information loss caused by the tangle between drift and useful information. Similar to the model-level methods like RMTL and TCTL, DCAE considers the discriminative information when correcting drift. Better yet, it can capture nonlinear and more abstract structures in data. The learned representation can be used in various prediction models, which makes it more convenient to use. The two DCAE variants outperform other methods especially when $n_{t,gas}$ is smaller.

More results of existing methods are listed in Table II. "No transfer" means the prediction model trained on batch 1 is directly applied on batches k ($k = 2, \ldots, 10$). The unsatisfactory performance proves the influence of drift. The two traditional methods based on component correction (CC-PCA and OSC) do not achieve large improvement because they rely on clear separation of drift and useful information in data. The results of ensemble, source domain adaptation extreme learning machine (DAELM-S), and manifold regularization with combination geodesic flow kernel (ML-comGFK) are copied from the original papers. DAELM-S achieves good results with the help of 30 selected labeled samples in each batch. Our proposed methods have a higher accuracy with only 10 transfer samples needed. The last three rows are results obtained by deep networks. The standard deviation of the average accuracy of the 10 runs is also calculated. In "joint train", samples in batch 1 and k are pooled together to train an stacked denoising autoencoder, as suggested by [25]. The poor performance indicates that this strategy is not suitable in drift correction. Among the results of DCAE-basic and DCAE-CL, the former method is better in earlier batches whereas the latter wins in latter batches which have larger drift. It indicates that DCAE-CL is preferable when the time-varying drift is more complex.

B. Breath analysis dataset

Disease screening and monitoring with e-noses is attracting more and more attention [2], [5]. Certain diseases can change the concentration of biomarkers in breath. For example, the concentration of breath acetone in diabetics is often higher than that of healthy people [5]. Therefore, it is possible to analyze a person's health state with an e-nose, which is non-invasive and convenient. To make this technology practical, instrumental variation and time-varying drift of e-noses have to be corrected. Drift correction algorithms based on labeled samples in target domains are not suitable in this problem, because collecting samples from patients is rather difficult. Therefore, only transfer-sample-based methods will be tested in this section.

A breath analysis dataset was collected by us in hospitals from 2014 to 2015. Two e-noses of the same model [5] were adopted for sample collection. Samples of five diseases

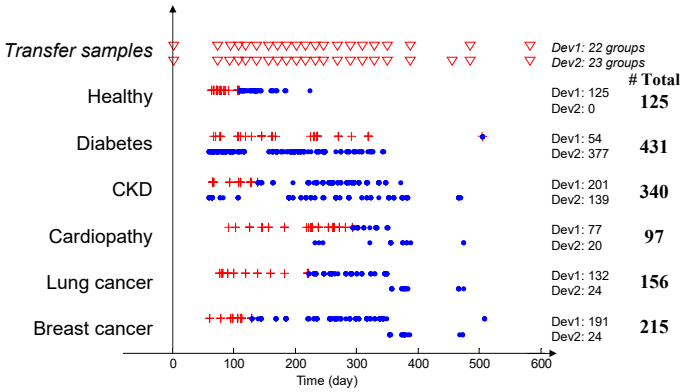

Fig. 5. Numbers (labeled on the right) and acquisition times (x-axis) of the samples in each class in the breath analysis dataset. The two rows of each class represent samples measured by the two devices. Each point denotes a breath sample. The red plus signs are training samples.

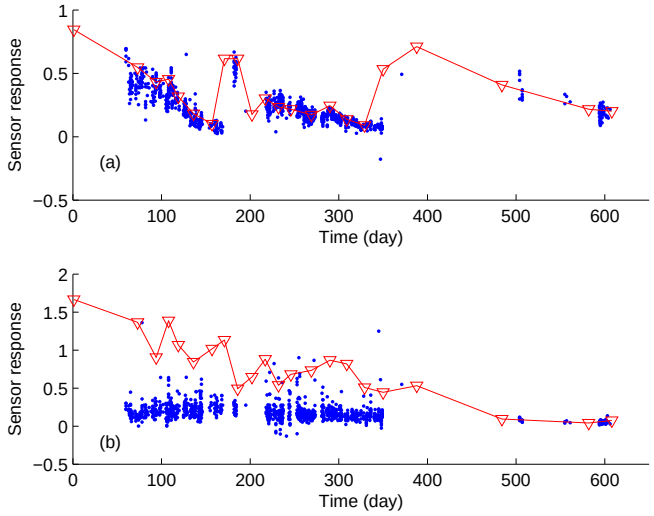

Fig. 6. Illustration of drift in the breath analysis dataset. Blue dots and red triangles represent breath and transfer samples, respectively. Plot (a) and (b) show the steady state responses of two sensors in device 1.

are selected for experiments in this paper, namely diabetes, chronical kidney disease (CKD), cardiopathy, lung cancer, and breast cancer. They have been proved to be related with certain breath biomarkers. The numbers and acquisition times of these samples and the healthy ones are demonstrated in Fig. 5. For the purpose of drift correction, transfer sample groups were also collected periodically with each e-nose, as shown in at the top of Fig. 5. Each group includes eight preselected standard gas samples. Fig. 6 illustrates the complex drift in this real-world dataset. In plot (a), the sensor suffered severe aging effect, so its response obviously decayed as time elapsed. We had to replace it with a new one on about day 200 and 400. In plot (b), the sensor aged slowly, but its responses to the transfer samples did not decay monotonously. The oscillation was caused by uncertainty during the collection of the transfer samples, which is common in weakly-controlled environments. Such noise in transfer samples will influence the drift correction process.

Five binary-class classification tasks (healthy vs. disease)

TABLE II
CLASSIFICATION ACCURACY (%) ON THE GAS SENSOR ARRAY DRIFT DATASET. BOLD VALUES INDICATE THE BEST RESULTS.

	Batch 2	3	4	5	6	7	8	9	10	Average
No transfer	88.59	66.96	40.99	54.82	43.22	44.40	31.63	45.74	39.11	50.61
CC-PCA [11]	90.92	40.86	47.20	59.39	56.74	56.71	36.39	45.32	37.72	52.36
OSC [12]	88.10	66.71	54.66	53.81	65.13	63.71	36.05	40.21	40.08	56.50
Ensemble [14]	74.36	87.83	**93.79**	95.43	69.17	69.72	91.84	76.38	65.50	80.45
ML-comGFK [17]	80.25	74.99	78.79	67.41	77.82	71.68	49.96	50.79	53.79	67.28
DAELM-S [15]	87.98	95.74	85.16	95.99	94.14	83.51	86.90	**100.0**	53.62	87.00
Joint train [25]	59.41	56.33	58.63	37.26	44.77	43.17	20.72	34.26	35.55	43.34±1.14
DCAE-basic	**97.58**	**96.82**	88.57	**98.73**	**95.40**	94.76	91.19	95.66	74.60	92.59±0.61
DCAE-CL	97.47	96.13	90.50	98.58	94.94	**95.47**	**92.62**	96.40	**76.82**	**93.21**±0.52

were carried out on this dataset. Because the number of samples in each class is different, the F-score was used as the accuracy criterion. In order to evaluate the drift correction algorithms, we chose the first 50 samples collected with device 1 in each class for training, as demonstrated in Fig. 5. Each sample was represented by the steady state responses of nine gas sensors. When a gas sensor is used to sense a gas sample, its response will reach a steady state in a few minutes. The steady state response has a close relationship with the concentration of the measured gas. Therefore, the 9D feature vector contains most information needed for disease screening. The features were preprocessed with the normalization method similar to that in Section V-A. Logistic regression was adopted as the final classifier.

The DCAE developed for this dataset has one hidden layer with four units. Because the numbers of features and classes are not large, further enlarging the network cannot improve the accuracy. Tanh activation function was used in the network including the correction layer. The domain features were created according to Eq. (4), where t was the exact acquisition time converted to years and the number of devices $n_{dev} = 2$. The impact of the hyper-parameters on the average accuracy is studied in Fig. 7. It is found that the change of accuracy is not large when λ_1 is varied. The best value of λ_2 is 2^{-6}, which is smaller than 2^{-3} in the gas sensor array drift dataset. We attribute it to the noisy transfer samples in this dataset. Larger λ_2 enforces the transfer samples to be aligned better in the representation space, which will cause overfitting if the transfer samples are noisy and their quantity is small. For h_{cor}, the accuracy is relatively stable when it is not smaller than 4. Finally we adopt $\lambda_1 = 2^{-4}, \lambda_2 = 2^{-6}$ and $h_{cor} = 4$.

The accuracy of various methods are compared in Table III. Recall that the tasks are distinguishing between healthy samples and samples with one kind of disease. If the prediction models trained on the original features are directly used for classification, the average accuracy is only 68.30%. Among all the tested methods, DCAE has the best performance. DCAE-CL outperforms DCAE-basic in all tasks. The last row of the table shows the results obtained by randomly selecting 50 training samples in each class. The experiment was repeated 20 times and the average accuracy is reported. No drift correction was done. Since the training samples were from all devices and time periods, the trained model should be robust to drift. The accuracy of DCAE-CL is close to random train.

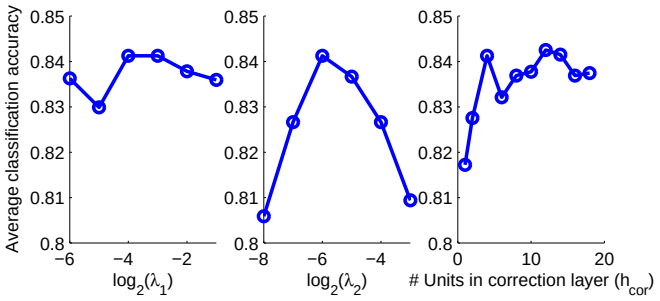

Fig. 7. Impact of the hyper-parameters on the average F-score of DCAE-CL in the breath analysis dataset.

TABLE III
CLASSIFICATION ACCURACY (%) ON THE BREATH ANALYSIS DATASET. BOLD VALUES INDICATE THE BEST DRIFT CORRECTION RESULTS.

	Task 1	2	3	4	5	Average
No transfer	63.20	71.49	75.87	64.36	66.55	68.30
Var. stdd.	52.13	49.58	65.85	47.88	45.47	52.18
CC-PCA [11]	70.82	79.24	84.59	74.19	76.24	77.02
Latent space	57.49	68.73	73.27	68.04	74.12	68.33±2.37
DCAE-basic	74.44	82.20	89.36	81.15	82.04	81.84±0.67
DCAE-CL	**82.16**	**84.27**	**89.94**	**81.34**	**82.92**	**84.13**±0.82
Random train	87.85	85.44	90.10	85.09	84.54	86.60

In the breath analysis dataset, the samples were collected in a stream. They drift in the data space continuously in both the training and the test set. TCTL and HHTL are designed for problems with discrete source and target domains, so they are not feasible in this dataset. CC-PCA and latent space correct all data as a whole. Latent space tries to align each group of transfer samples to the reference group, which is not suitable when the transfer samples are noisy. This could be the reason why its performance is not good. We also tried to split the data into batches so that each batch had a group of transfer samples. A breath sample was assigned to the batch whose transfer sample group was closest to it in time. Then, variable standardization was applied for drift correction. The results are

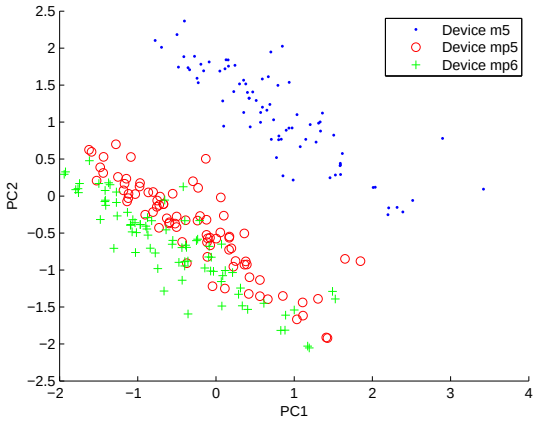

Fig. 8. Scatter of the samples measured with the three devices in the corn dataset.

listed in the row "var. stdd.". Its poor performance can also be caused by the noise in transfer samples, which makes the correction inaccurate. On the contrary, the correction output in DCAE is smooth over time as long as λ_2 is not too large. Two samples will be given similar correction output if their acquisition times are close. This is consistent with the prior knowledge that the time-varying drift usually changes slowly in time, which helps to smooth the noise in transfer samples. Besides, by using the domain features, DCAE can correct continuous drift intrinsically and without loss of the continuity information.

C. Corn dataset

Spectrometer data and e-nose data share a lot of common property and drift correction algorithms in both areas often work universally. In this section, we test our methods on a spectroscopy dataset with instrumental variation. The corn dataset[2] was collected with three near-infrared spectrometers designated as m5, mp5, and mp6. Each device was adopted to measure the moisture, oil, protein, and starch contents of 80 corn samples. The ranges of the measured values are [9.377, 10.993], [3.088, 3.832], [7.654, 9.711], and [62.826, 66.472], respectively. Each sample is represented by a spectra with 700 features. The features were preprocessed with the normalization method similar to that in Section V-A. One can observe the discrepancy among the three devices from Fig. 8. We follow the experimental setting in [16] and study the calibration transfer from m5 to the other two devices. Calibration transfer is the term used in machine olfaction and spectroscopy for transferring the model of one device to another. A four-fold cross validation was made by assigning every fourth sample to the test set. Regression models were trained on the training samples collected with m5 and tested on the test samples collected with mp5 and mp6. Linear ridge regression was adopted as the final regression algorithm. Root mean square error (RMSE) was used as the accuracy criterion.

DCAE-basic was used for this dataset since it contains no time-varying drift. Considering that the complexity of the dataset is relatively small, we adopted a network with one hidden layer and linear activation function. The number of units in the hidden layer was 15. The domain feature vectors were created according to the one-hot coding scheme in Section III-B with $n_{dev} = 3$. Because the four values to predict have different variances, we normalized them to zero mean and unit variance before training DCAE, then transformed them back when computing RMSE. For this dataset, we found that small RMSE was obtained when setting $\lambda_1 = 2^1, \lambda_2 = 2^4$. The optimal hyper-parameters are larger than the previous datasets, which is because there is much less noise in this dataset. It is less prone to overfitting when the supervised loss and alignment error are given larger weights.

Results of various methods are compared in Table IV. For each calibration transfer algorithm except RMTL, LLR was used to select seven transfer samples (or labeled target samples) from the training data. The results of RMTL are copied from [16], which only provided the results on mp6. The "train on target" method trained and tested regression models on the same target device. Its results can be regarded as goals for calibration transfer. It can be found that TCTL and DCAE-basic achieved the lowest average RMSE for calibration transfer. Their superiority comes from the fact that they jointly optimize the supervised loss and the alignment error when learning models or representations.

D. Impact of different training procedures

According to Algorithm III.1, the weights in DCAE are learned in three steps: unsupervised pretraining (unsup.), supervised fine-tuning (sup.), and final optimization of the objective function (obj.), Eq. (7). In order to explore the necessity of the two steps before the final optimization, we compare four different training procedures in Table V. The number after the name of a step indicates its preset maximum iteration number. When one step is omitted, the maximum iteration number of the final optimization is increased for fair comparison.

As shown in the table, the proposed procedure achieves the best results in all three datasets. On the other side, directly optimizing the objective function is nearly always the worst one. Similar to the pretraining step in original autoencoders, the two steps before the final optimization drive the weights to approach a "good" local minimum, where "good" is in terms of generalization error [21]. Nevertheless, the weights obtained by directly optimizing Eq. (7) may have larger generalization error, even if it can produce a smaller value in the objective function. Besides, the actual iteration number of the proposed procedure is also the smallest among the four procedures, because both unsupervised pretraining and supervised fine-tuning are easier tasks than minimizing Eq. (7) and converge faster. In the experiment for batches 1 and 2 in the first dataset, the time costs for optimizing the three steps are 1.7, 7.3, and 48.1s, respectively.

VI. Conclusion

In this paper, we proposed drift correction autoencoder (DCAE) for joint representation learning and drift correction

[2]http://www.eigenvector.com/data/Corn/

TABLE IV
REGRESSION RMSE ON THE CORN DATASET. BOLD VALUES INDICATE THE BEST CALIBRATION TRANSFER RESULTS.

	Mp5 as target domain					Mp6 as target domain				
	Moisture	Oil	Protein	Starch	Average	Moisture	Oil	Protein	Starch	Average
No transfer	1.499	0.288	1.097	2.086	1.242	1.632	0.343	1.350	2.063	1.347
Var. stdd. [10]	0.269	0.106	0.147	0.354	0.219	0.303	0.110	0.153	0.356	0.231
DAELM-S [15]	0.238	0.109	0.150	0.365	0.215±0.001	0.266	0.113	0.154	0.368	0.225±0.002
RMTL [16]			–			0.235	0.089	0.160	0.357	0.210
TCTL [18]	**0.182**	0.104	0.148	0.351	0.196	**0.193**	0.104	0.143	**0.337**	**0.194**
Latent space	0.191	0.095	0.162	0.370	0.205±0.002	0.197	0.097	0.166	0.385	0.211±0.001
HHTL [26]	0.262	**0.092**	**0.146**	0.404	0.226	0.300	**0.091**	**0.140**	0.377	0.227
DCAE-basic	0.183	0.096	0.149	**0.348**	**0.194**±0.002	0.199	0.096	0.148	0.350	0.198±0.002
Train on target	0.137	0.101	0.140	0.363	0.185	0.139	0.099	0.149	0.369	0.189

TABLE V
CLASSIFICATION ACCURACY (%) OR RMSE ON THE THREE DATASETS WITH DIFFERENT TRAINING PROCEDURES. BOLD VALUES INDICATE THE BEST RESULTS.

Procedure	Dataset 1	Dataset 2	Dataset 3 (RMSE)
Obj.(3000)	76.56±2.36	81.28±2.11	0.2096±0.0014
Unsup.(1000) + obj.(2000)	75.83±2.15	83.41±2.94	0.2055±0.0014
Sup.(1000) + obj.(2000)	93.04±0.44	83.44±0.89	0.1964±0.0017
Unsup.(1000) + sup.(1000) + obj.(1000)	**93.21**±0.52	**84.13**±0.82	**0.1961**±0.0022

in machine olfaction. The main difference between DCAE and the original stacked autoencoder lies in three aspects: 1) Domain features and the correction layer are introduced in DCAE to explicitly model instrumental variation and time-varying drift; 2) The component of drift is removed in the encoding process and recovered in the decoding one, so that the hidden representation of DCAE is drift-corrected; 3) Transfer samples are utilized to learn the correction weight matrices. Compared with other drift correction algorithms, DCAE has several characteristics: 1) It tackles both discrete and continuous drift in the training and test data naturally without having to split the data into different domains; 2) The correction output in DCAE is intrinsically smooth over time, which is consistent with the temporal smoothness prior of the time-varying drift, so that the influence of noisy transfer samples can be diminished; 3) When correcting drift, DCAE keeps the learned representation to be representative and discriminative as well at the cost of two regularization hyper-parameters. According to our experiments, they can be set according to the noise level of the dataset and the transfer samples.

The superiority of DCAE is more obvious when the drift in the dataset is complex (e.g. Section V-A and V-B). Besides, when the size of the dataset is not large (which is common in machine olfaction), it may be better to use a smaller network. The application scope of DCAE may be further extended. By simply modifying the domain features, other influential factors such as temperature and humidity can also be corrected as long as suitable transfer samples are collected. Apart from machine olfaction, it is also viable in fields such as spectroscopy, where data measured by sensors or devices contain drift and transfer samples can be collected. Future study may include deeper exploiting the information in the unlabeled samples to improve the effect of correction.

ACKNOWLEDGMENT

The authors would like to thank the providers of the datasets used in this paper. The authors would also like to thank Lu Kou for his help.

REFERENCES

[1] J. W. Gardner and P. N. Bartlett, "A brief history of electronic noses," *Sens. Actuators B: Chem.*, vol. 18, no. 1, pp. 210–211, 1994.
[2] F. Röck, N. Barsan, and U. Weimar, "Electronic nose: current status and future trends," *Chem. Rev.*, vol. 108, no. 2, pp. 705–725, 2008.
[3] K. Brudzewski, S. Osowski, and A. Dwulit, "Recognition of coffee using differential electronic nose," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 6, pp. 1803–1810, 2012.
[4] L. Zhang, F. Tian, C. Kadri, B. Xiao, H. Li, L. Pan, and H. Zhou, "On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality," *Sens. Actuators: B. Chem.*, vol. 160, no. 1, pp. 899–909, 2011.
[5] K. Yan, D. Zhang, D. Wu, H. Wei, and G. Lu, "Design of a breath analysis system for diabetes screening and blood glucose level prediction," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 11, pp. 2787–2795, 2014.
[6] S. Marco and A. Gutiérrez-Gálvez, "Signal and data processing for machine olfaction and chemical sensing: a review," *IEEE Sens. J.*, vol. 12, no. 11, pp. 3189–3214, 2012.
[7] S. Di Carlo and M. Falasconi, *Drift correction methods for gas chemical sensors in artificial olfaction systems: techniques and challenges*. InTech, 2012, ch. 14, pp. 305–326.
[8] R. N. Feudale, N. A. Woody, H. Tan, A. J. Myles, S. D. Brown, and J. Ferré, "Transfer of multivariate calibration models: a review," *Chemometr. Intell. Lab.*, vol. 64, no. 2, pp. 181–192, 2002.
[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
[10] K. Yan and D. Zhang, "Improving the transfer ability of prediction models for electronic noses," *Sens. Actuators B: Chem.*, vol. 220, pp. 115–124, 2015.
[11] T. Artursson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, and M. Holmberg, "Drift correction for gas sensors using multivariate methods," *J. Chemometr.*, vol. 14, no. 5-6, pp. 711–723, 2000.
[12] M. Padilla, A. Perera, I. Montoliu, A. Chaudry, K. Persaud, and S. Marco, "Drift compensation of gas sensor array data by orthogonal signal correction," *Chemometr. Intell. Lab.*, vol. 100, no. 1, pp. 28–35, 2010.

[13] A.-C. Romain and J. Nicolas, "Long term stability of metal oxide-based gas sensors for e-nose environmental applications: An overview," *Sens. Actuators B: Chem.*, vol. 146, no. 2, pp. 502–506, 2010.
[14] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sens. Actuators B: Chem.*, vol. 166, pp. 320–329, 2012.
[15] L. Zhang and D. Zhang, "Domain adaptation extreme learning machines for drift compensation in e-nose systems," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1790 – 1801, 2015.
[16] Y. Binfeng and J. Haibo, "Near-infrared calibration transfer via support vector machine and transfer learning," *Analytical Methods*, vol. 7, no. 6, pp. 2714–2725, 2015.
[17] Q. Liu, X. Li, M. Ye, S. S. Ge, and X. Du, "Drift compensation for electronic nose by semi-supervised domain adaption," *IEEE Sens. J.*, vol. 14, no. 3, pp. 657–665, 2014.
[18] K. Yan and D. Zhang, "Calibration transfer and drift compensation of e-noses via coupled task learning," *Sens. Actuators B: Chem.*, vol. 225, pp. 288–297, 2016.
[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.
[20] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *Unsupervised and Transfer Learning Challenges in Machine Learning*, vol. 7, p. 19, 2012.
[21] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.
[22] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recogn. Lett.*, vol. 42, pp. 11–24, 2014.
[23] M. Längkvist and A. Loutfi, "Unsupervised feature learning for electronic nose data applied to bacteria identification in blood," in *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
[24] M. Längkvist, S. Coradeschi, A. Loutfi, and J. B. B. Rayappan, "Fast classification of meat spoilage markers using nanostructured zno thin films and unsupervised feature learning," *Sensors*, vol. 13, no. 2, pp. 1578–1592, 2013.
[25] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 513–520.
[26] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
[27] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: transfer learning with deep autoencoders," in *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 4119–4125.
[28] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1068–1072, 2014.
[29] C. Kandaswamy, L. M. Silva, L. A. Alexandre, J. M. Santos, and J. M. de Sá, "Improving deep neural network performance by reusing features trained with transductive transference," in *Artificial Neural Networks and Machine Learning–ICANN 2014*. Springer, 2014, pp. 265–272.
[30] S. Chopra, S. Balakrishnan, and R. Gopalan, "Dlid: Deep learning for domain adaptation by interpolating between domains," in *ICML workshop on challenges in representation learning*, vol. 2, 2013, p. 5.
[31] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.
[32] M. Kan, S. Shan, H. Chang, and X. Chen, "Stacked progressive auto-encoders (spae) for face recognition across poses," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
[33] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang, "Active learning based on locally linear reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2026–2038, 2011.
[34] J. Chou, *Gas Sensor Calibration*, 1st ed. McGraw-Hill Professional Publishing, 1999, ch. 11, pp. 161–173.
[35] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a cpu and gpu math expression compiler," in *Proceedings of the Python for scientific computing conference (SciPy)*, vol. 4. Austin, TX, 2010, p. 3.