Multimodal Vigilance Estimation with Adversarial Domain Adaptation Networks

He Li, Wei-Long Zheng, Bao-Liang Lu*
Center for Brain-like Computing and Machine Intelligence
Department of Computer Science and Engineering
Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, 200240, China

Abstract—**Robust vigilance estimation during driving is very crucial in preventing traffic accidents. Many approaches have been proposed for vigilance estimation. However, most of the approaches require collecting subject-specific labeled data for calibration which is high-cost for real-world applications. To solve this problem, domain adaptation methods can be used to align distributions of source subject features (source domain) and new subject features (target domain). By reusing existing data from other subjects, no labeled data of new subjects is required to train models. In this paper, our goal is to apply adversarial domain adaptation networks to cross-subject vigilance estimation. We adopt two kinds of recently proposed adversarial domain adaptation networks and compare their performance with those of several traditional domain adaptation methods and the baseline without domain adaptation. A publicly available dataset, SEED-VIG, is used to evaluate the methods. The dataset includes electroencephalography (EEG) and electrooculography (EOG) signals, as well as the corresponding vigilance level annotations during simulated driving. Compared with the baseline, both adversarial domain adaptation networks achieve improvements over 10% in terms of Pearson's correlation coefficient. In addition, both methods considerably outperform the traditional domain adaptation methods.**

Index Terms—**adversarial network, domain adaptation, electroencephalography (EEG), electrooculography (EOG), vigilance estimation.**

I. Introduction

The high incidence of traffic accidents has always been a very serious problem. Absence of vigilance is believed to be one of the most significant factors that cause traffic accidents. According to the government report, there were 396,000 drowsy driving related traffic crashes from 2011 to 2015 in the USA [1]. When people are in states of low vigilance levels, their abilities of handling accidental events weaken, and they are liable to cause accidents. As a consequence, vigilance estimation during driving is of great importance.

During the last several years, there has been plenty of progress in measuring mental and physical states with *Electroencephalography* (EEG) signals [2]–[4]. EEG signals are time series signals obtained by recording brain electromagnetic fields. Moreover, multimodal approaches for mental and physical applications have been developed in recent years [5]–[7]. Signals from different modalities are related to different aspects of subject states. Thus, integrations of features from different modalities help to form a robust and powerful system. Among all of the applications using multimodal integration, vigilance estimation is one of the very interesting topics. Various studies on this topic have been reported [8]–[10]. However, most of them only focus on classifying driver states to some predefined vigilance level categories. The more desirable system should be able to output continuous estimates as the vigilance levels in real time. In our previous work, a simulated driving system was developed in which vigilance levels of drivers were estimated, and the vigilance estimation was improved by using multimodal approaches and incorporating temporal dependency [11]–[13].

For two data sets drawn from different but related distributions (source domain and target domain), *domain adaptation* methods can be used to enhance the performance of models trained on the source domain and tested on the target domain. As a result, domain adaptation plays an important role in saving manpower and material resources by reusing existing data from relevant domains for model training so that few or no labeled data from the target domain is required. Moreover, recent years have seen a great boom in the field of *deep adversarial networks* [14]. There are several domain adaptation methods proposed using deep adversarial networks that achieve state of the art performance in object recognition problems [15], [16]. The basic idea of these methods is to adversarially train feature extractors and domain classifiers so that the feature extractors can extract domain invariant features. In this paper, we refer to this kind of neural networks as *adversarial domain adaptation networks*.

Because collecting labeled data for vigilance estimation could be high-cost, domain adaptation for cross-subject vigilance estimation has become very important. However, to the best of our knowledge, domain adaptation for cross-subject vigilance estimation has not been fully investigated yet. In our previous work [17], *transfer component analysis* [18] was used for the classification problem of cross-subject fatigue detection. In addition, various domain adaptation methods have been proposed for cross-subject vigilance estimation [19]–[21], but these approaches require some labeled data of new subjects for calibration. In this paper, we integrate the multimodal approach [11] and adversarial domain adaptation

*Corresponding author: Bao-Liang Lu (bllu@sjtu.edu.cn) networks [15], [16] to build cross-subject vigilance estimation models without using any label information from new subjects. We examine several popular domain adaptation methods and make a systematic comparison on their performance. The experimental results indicate that the adversarial domain adaptation networks considerably improve the vigilance estimation accuracy.

The rest part of this paper is organized as follows. Section II introduces different domain adaptation methods used in this paper. Section III describes the multimodal dataset used in this paper and the corresponding data preprocessing procedure. Section IV presents the domain adaptation results. Section V provides the conclusions of this paper.

II. Domain Adaptation Methods

A. Basic Idea

Domain adaptation is a branch of transfer learning (i.e., transductive learning within the same feature space [22]). The source domain is denoted by $D_s = \{X_s, Y_s\}$, in which $X_s = \{\boldsymbol{x}_{s_1}, \boldsymbol{x}_{s_2}, \cdots, \boldsymbol{x}_{s_n}\}$ is the input and $Y_s = \{y_{s_1}, y_{s_2}, \cdots, y_{s_n}\}$ is the corresponding label set. The values of X_s and Y_s are drawn from the joint distribution $P(X_s, Y_s)$. Similarly, the target domain denoted by $D_t = \{X_t, Y_t\}$ corresponds to data and labels drawn from the joint distribution $P(X_t, Y_t)$. In this paper, we consider unsupervised domain adaptation, which means label information from the target domain is not required. Typically, the marginal distributions of the input data are different between source domain and target domain: $P(X_s) \neq P(X_t)$. This is usually referred to as domain shift and is considered to be the key problem that leads to poor performance when a model is trained and tested on data from different domains. To eliminate the influence of domain shift, feature-based domain adaptation methods try to find a proper transformation function $\boldsymbol{\phi}(\cdot)$ that aligns the data into a new feature space where $P(\boldsymbol{\phi}(X_s)) \approx P(\boldsymbol{\phi}(X_t))$.

B. Traditional Methods

Several traditional feature-based domain adaptation methods are briefly introduced below, and their performance on cross-subject vigilance estimation will be compared in section IV.

1) Geodesic Flow Kernel (GFK) [23]*:* In GFK, the *Principal component analysis* (PCA) bases of source and target domains are first computed. The two sets of bases are then regarded as two points ($\boldsymbol{P}_s, \boldsymbol{P}_t \in \mathbb{R}^{D\times d}$, each column indicates a base vector) on a *Grassmannian*. After that, a geodesic flow $\boldsymbol{\Phi}(t)$ is defined with $t \in [0, 1]$ under the constraints $\boldsymbol{\Phi}(0) = \boldsymbol{P}_s$ and $\boldsymbol{\Phi}(1) = \boldsymbol{P}_t$. So the geodesic flow can be interpreted as a path from the source domain bases to the target domain bases. The projections of two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with all of the bases on that path (an infinite number of them) can be concatenated to form infinite-dimensional vectors $\boldsymbol{z}_i^\infty$ and $\boldsymbol{z}_j^\infty$. With the help of kernel trick, the product of the two infinite-dimensional vectors can be obtained easily. This kind of products is then used for model training. GFK aims to find an intermediate space where the idiosyncrasies in both domains are reduced while the common idiosyncrasies are preserved.

2) Subspace Alignment (SA) [24]*:* Similar to GFK, SA also utilized PCA bases to generate corresponding subspaces for source and target domains. However, instead of finding an intermediate space for the two domains, SA directly aligns the source domain subspace to the target one by a linear transformation represented by M. Borrowing the notations in Section II-B1, M is obtained by minimizing the *Frobenius norm* $||\boldsymbol{P}_s M - \boldsymbol{P}_t||_F^2$. After that, $\boldsymbol{P}_t$ is used for projecting target domain data and $\boldsymbol{P}_s M$ is used for projecting source domain data to their subspaces, respectively. The projected features are then used for training models.

3) Transfer Component Analysis (TCA) [18]*: Maximum mean discrepancy* (MMD) [25] is broadly used in domain adaptation methods as a metric of distribution discrepancies and is defined as the squared distance between the kernel embeddings of the source and target data in a *reproducing kernel Hilbert space* (RKHS). TCA aims to find a projection to a new space where MMD between source domain and target domain data is minimized. It works by solving the following constrained optimizing problem:

$$\begin{aligned} \min_{W} \quad & \mathrm{tr}(W^\top KLKW) + \mu \mathrm{tr}(W^\top W), \\ \text{s.t.} \quad & W^\top KHKW = I, \end{aligned} \tag{1}$$

where W is equivalent to a projection matrix, K is the kernel matrix defined on all the data, L is the coefficient matrix, H is the centering matrix, I is an identity matrix, and μ is a tradeoff parameter. The first term corresponds to MMD between source and target domain embeddings in a RKHS. The second term controls the complexity of the embedding. The restriction term helps to preserve the data variance in the projected space. In addition, semi-supervised transfer component analysis (SSTCA) is an extension to TCA that takes label information into consideration when finding the projection.

4) Maximum Independence Domain Adaptation (MIDA) [26]*:* MIDA is a recently developed method, in which a domain feature vector $\boldsymbol{d} \in \mathbb{R}^{m_d}$ for each original feature vector $\boldsymbol{x}$ is constructed. Each element d_i represents some domain information. In the simplest way, d_i takes the value 1 if the corresponding $\boldsymbol{x}$ is from the ith domain and 0 otherwise. Then each data point is replaced by the augmentation of the original features and domain features (i.e. $\hat{\boldsymbol{x}} = [\boldsymbol{x}^\top, \boldsymbol{d}^\top]^\top \in \mathbb{R}^{m+m_d}$). MIDA tries to find a domain-invariant subspace of the augmented features in which the projected features are independent to the original domain features. To measure the dependency between projected features and domain features, *Hilbert-Schmidt independence criterion* (HSIC) is adopted [27]. The learning problem of MIDA can be expressed as

$$\begin{aligned} \max_{W} \quad & -\mathrm{tr}(K_{\boldsymbol{z}} H K_{\boldsymbol{d}} H) + \mu \mathrm{tr}(W^\top K_{\hat{\boldsymbol{x}}} H K_{\hat{\boldsymbol{x}}} W), \\ \text{s.t.} \quad & W^\top W = I, \end{aligned} \tag{2}$$

where W is equivalent to a projection matrix, $K_{\hat{\boldsymbol{x}}}$, $K_{\boldsymbol{z}}$, and $K_{\boldsymbol{d}}$ are, respectively, the kernel matrices defined on the concatenated features, subspace of the concatenated features defined by W, and the domain features, H is the centering matrix, I is an identity matrix, and μ is a tradeoff parameter. The first term corresponds to the HSIC, the second term helps to preserve data variance, W is a projection matrix and its scale is constrained by the restriction term. Similar to TCA, MIDA also has a semi-supervised version (SSMIDA).

C. Adversarial Network Methods

Two kinds of adversarial domain adaptation networks are described here. Their performance is compared with those of the traditional domain adaptation methods in section IV.

1) Domain-Adversarial Neural Network (DANN) [15]*:* DANN was first proposed in [15], and its properties and applications are then further explored in [28]. The model can be divided into the following three parts: a feature extractor G_f, a label predictor G_y, and a domain classifier G_d. There exist adversarial relationships between the feature extractor and the domain classifier. The feature extractor, as the name implies, extracts new features from input features: $\boldsymbol{f} = G_f(\boldsymbol{x}; \boldsymbol{\theta}_f)$. Here $\boldsymbol{x}$ denotes input feature vector and $\boldsymbol{f}$ denotes the corresponding output feature vector in a new feature space. The outputs are then fed into the label predictor and the domain classifier. The label predictor provides predictions of the corresponding labels: $\hat{y} = G_y(\boldsymbol{f}; \boldsymbol{\theta}_y)$. The domain classifier distinguishes which domain the input is from: $\hat{d} = G_d(\boldsymbol{f}; \boldsymbol{\theta}_d)$. The three parts are updated simultaneously with the objective function:

$$E(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \boldsymbol{\theta}_d) = \sum_{i=1}^{N} L_y(\hat{y}_i, y_i) - \lambda \sum_{i=1}^{N} L_d(\hat{d}_i, d_i), \qquad (3)$$

where the first term $L_y(\cdot,\cdot)$ is the loss for label prediction, and $L_d(\cdot,\cdot)$ corresponds to the loss for domain classification. The update rule is designed as follows:

$$\begin{aligned} (\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y) &= \arg\min_{\boldsymbol{\theta}_f, \boldsymbol{\theta}_y} E(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \hat{\boldsymbol{\theta}}_d), \\ \hat{\boldsymbol{\theta}}_d &= \arg\max_{\boldsymbol{\theta}_d} E(\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y, \boldsymbol{\theta}_d). \end{aligned} \qquad (4)$$

It can be observed that the label predictor and domain classifier are trained so that the corresponding losses are minimized. The feature extractor is trained so that the label prediction loss is minimized while the domain classification loss is maximized. So the feature extractor is trying to extract features that are good for label prediction, but not easy to distinguish which domain the features come from. In this way, the feature extractor is to extract domain invariant features, so the domain shift can be eliminated.

2) Adversarial Discriminative Domain Adaptation (ADDA) [16]*:* Similar to DANN, ADDA also can be divided into three parts, except that there are two feature extractors, one for source domain data and another for target domain data. Let G_{f_0} and G_{f_1} be the corresponding feature extractors for source domain and target domain, respectively. The training procedure is two-stage. In the first stage, G_{f_0} and the label predictor G_y are trained with source domain data so that the prediction loss is minimized. After the training, the parameters of G_{f_0} and G_y are fixed during the following process. In the second stage, G_{f_1} is initialized with the parameters of G_{f_0}. Then G_{f_1} and G_d are trained adversarially: G_d is trained to discriminate source domain data and target domain data, while G_{f_1} is trained to fool G_d. So, after the training, the feature extractor G_{f_1} aligns the distribution of the target domain data to that of the source domain data.

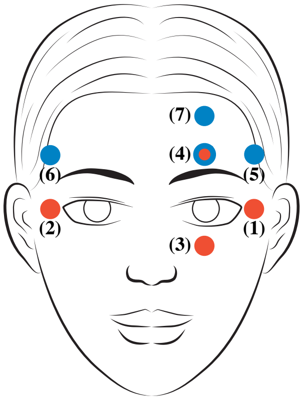


Fig. 1: The placement of electrodes. The red points 1–4 indicate the position of traditional EOG electrode placement setup while the blue points 4-7 indicate the forehead setup. Point 4 is shared by both of the setups.

III. Data Description and Preprocessing

A. The SEED-VIG Dataset

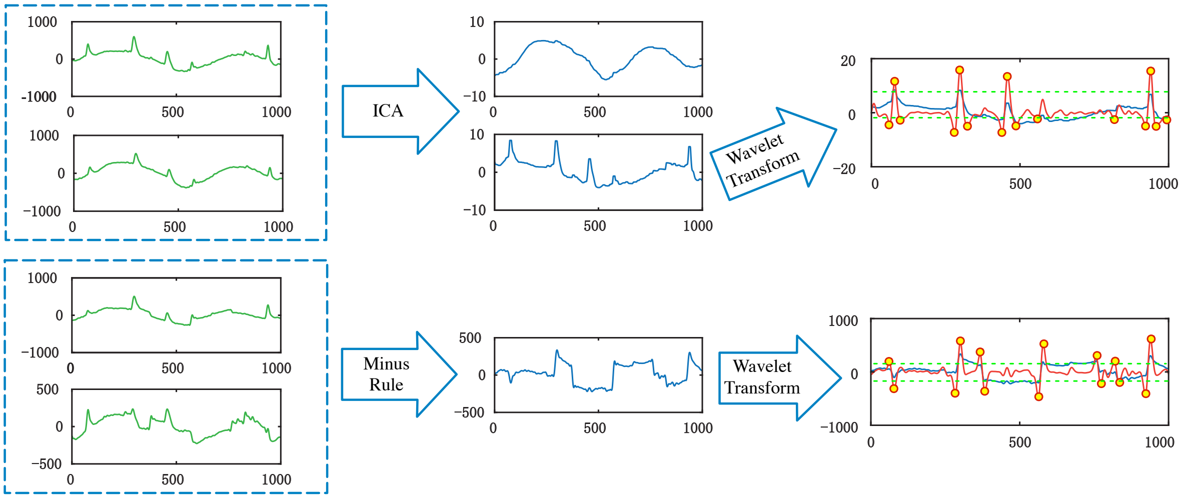
To obtain vigilance changing data of subjects during driving, a simulated driving system was developed [11]. The system is composed of a large LCD screen, a real vehicle, and a software controller. Animation shown on the screen is simultaneously updated according to operations of subjects. The operations include steering, throttle controlling and braking and lead the subject to feel like driving on a real highway.

In the simulated driving experiments, 23 volunteers (their average age is 23.3 years old, 12 of them being females) were selected as subjects. All of the subjects own normal or corrected-to-normal vision. Drugs affecting nervous system were prohibited before the experiments. The subjects were required to attend the experiments during early afternoons or late nights to arouse fatigue easily. The experiments lasted for 2 hours, during which data were recorded. However, the first and the last 60-second data were discarded to avoid external influences. The SEED-VIG dataset used in this paper is publicly available[1].

Both EEG and *electrooculography* (EOG) signals are recorded using the Neuroscan system at the sampling rate of 1000 Hz. The corresponding electrodes were placed according to the forehead placement [29] as shown in Figure 1.

EEG signals from posterior and temporal sites were also recorded. But only forehead EEG and EOG signals are used here for two reasons: a) it is relatively easier to implement the forehead placement in real-world wearable devices; and b)

[1] The SEED-VIG dataset: http://bcmi.sjtu.edu.cn/~seed/download.html



(a) Original Signal

(b) VEO and HEO

(c) Peak Detection

Fig. 2: EOG signal extraction and EOG feature extraction. The plots shown in Figure 2(a) are original signals recorded by the four forehead electrodes within one of the 8-second time window (each of them from top to bottom corresponds to electrodes 4, 7, 5, and 6 in Figure 1, respectively). Because the signals are downsampled to 125Hz, there are 1000 samples for each channel. The upper two figures in Figure 2(b) show the ICA components extracted from electrodes 4 and 7 as the arrow shows. It can be observed that the lower figure corresponds to the VEO component while the upper one corresponds to some background signals. Each spike in the figure corresponds to a blink event. The bottom figure in Figure 2(b) shows the result of applying minus rule to the time series recorded by electrodes 5 and 6 as the arrow shows. The up-rising edges and down-falling edges correspond to saccade events. Figure 2(c) shows the result by applying Mexican hat wavelet to the VEO and HEO signals. The yellow points are the results of peak detection.

with the forehead placement, the vigilance estimation models can achieve comparable performance [11].

B. Feature Extraction

1) Data Preprocessing: The raw data recorded by the forehead electrodes were firstly downsampled to 125 Hz. Then the signals were segmented to epochs by 8-second non-overlapping time windows. The features extracted from each epoch represent one input vector. For each subject, because there are $7200 - 60 \times 2 = 7080$ seconds of valid raw signals, there are $7080/8 = 885$ inputs.

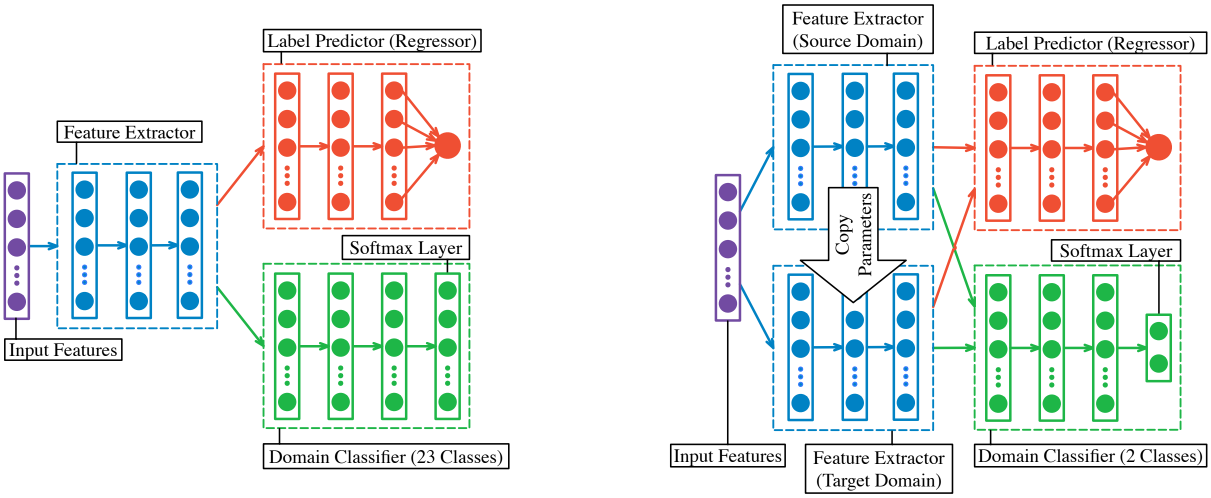
2) EOG Signal Extraction: Two important EOG components are horizontal EOG (HEO) and vertical EOG (VEO). As is shown in Figure 1, for traditional EOG placement setup, HEO and VEO are obtained by subtracting electrodes 1 from 2, and electrodes 3 from 4, respectively. For the forehead EOG placement setup, subtraction was made between electrodes 5 and 6 to obtain HEO$_f$. While, to obtain VEO$_f$, it is better to apply *independent component analysis* (ICA) [30] on the signals from electrodes 4 and 7 and select the EOG component, as it was verified in [11]. So we have $\text{HEO}_f = \boldsymbol{e}_5 - \boldsymbol{e}_6$ and $\text{VEO}_f = \text{ICA}(\boldsymbol{e}_4, \boldsymbol{e}_7)$. For concision of expression, the 'f' subscripts will be omitted in the rest of this paper. The extraction result of one epoch is shown in Figure 2.

3) EOG Feature Extraction: EOG feature extraction consists in edge detection for HEO and VEO signals. Rising edges and falling edges within HEO signals correspond to eye saccade events while rising edges with immediately following falling edges within VEO signals correspond to eye blink events. To achieve reliable edge detection, the wavelet transform method was used as introduced in [31]. Both HEO and VEO signals were processed with Mexican hat wavelet at the scale of 8. As is shown in Figure 2, the edge detection problem then became peak detection problem. With robust peak detection methods, eye movement (blink and saccade) information can be extracted effectively. After that, a total number of 36 features were generated according to the extracted eye movement information. A detailed list of the 36 features is shown in Table I.

TABLE I
36 Features Extracted from EOG Signals

Source	Features
Blink	Maximum/mean/sum of blink rate maximum/minimum/mean of blink amplitude, mean/maximum of blink rate variance and amplitude variance power/mean power of blink amplitude blink numbers
Saccade	Maximum/minimum/mean of saccade rate and saccade amplitude, maximum/mean of saccade rate variance and amplitude variance, power/mean power of saccade amplitude, saccade numbers
Fixation	Mean/maximum of blink duration variance and saccade duration variance maximum/minimum/ mean of blink duration and saccade duration

4) EEG Signal Extraction: Similar to the procedure of EOG signal extraction, ICA was also applied to EEG signal extraction. If the unmixing equation of ICA is $U = WE$, then the reconstruction of EEG signals for the forehead channels will be $\widetilde{E} = W^{-1}\widetilde{I}U$, where E indicates the raw signals from forehead electrodes 4~7, $\widetilde{E}$ indicates the reconstructed forehead EEG signals and $\widetilde{I}$ indicates a modified identity matrix with diagonal elements corresponding to EOG components set to zeros.


Label Predictor (Regressor)
Feature Extractor
Softmax Layer
Input Features
Domain Classifier (23 Classes)
Feature Extractor (Source Domain)
Label Predictor (Regressor)
Copy Parameters
Softmax Layer
Input Features
Feature Extractor (Target Domain)
Domain Classifier (2 Classes)

(a) Network structure of DANN.

(b) Network structure of ADDA.

Fig. 3: The network structures of adversarial network methods used in this paper.

5) EEG Feature Extraction: Differential entropy (DE) features were extracted from each epoch of $\widetilde{E}$. The frequency bands were 2 Hz bands from 1 Hz to 50 Hz (i.e., frequency bands of 1~2 Hz, 2~4 Hz, ..., 48~50 Hz). So the number of dimensions for EEG features is $4 \times 25 = 100$.

6) Feature Fusion and Smoothness: To take advantage of the multimodal features, feature fusion was applied by concatenating the 36 EOG features and 100 EEG features. So a 136-dimension feature vector was generated for each epoch. After that, to reduce the influence of artifacts, feature vectors were smoothed in sequential order by the moving average algorithm with the window size of 30.

C. Vigilance Annotation

Eye tracking glasses were used to obtain PERCLOS indices [32] of the subjects during the driving experiment. The name 'PERCLOS' is the abbreviation for 'percentage of eye closure', so its value ranges from 0 (high vigilance level) to 1 (low vigilance level). The PERCLOS index values were further smoothed with the moving average algorithm as the vigilance annotations.

IV. Domain Adaptation Results and Discussion

A. Domain Adaption Settings

We have extracted 885 feature vectors for each of the 23 subjects. Each feature vector is attached with the corresponding vigilance annotation. Our objective now is to perform domain adaptation between different subjects. The leave-one-subject-out cross-validation algorithm is applied, which means, for each domain adaptation method there are a few runs, and for each run the data from one of the subjects are regarded as target domain while the data from other subjects as source domains.

All the domain adaptation methods introduced in Section II are adopted. Besides, the baseline results are obtained by directly using the features without any domain adaptation. For the traditional domain adaptation methods and the baseline method, the linear kernel *support vector regression* (SVR) algorithm [33] is used for the regressors. For TCA and MIDA, both of the unsupervised and semi-supervised versions are adopted. Because TCA, SA, GFK, and ADDA can not be directly applied to multiple source domains, all source domain data (i.e., data of 22 subjects) are concatenated as data of one source domain for these methods. *Multi-layer perceptrons* (MLPs) are used for the feature extractors, the label predictors, and the domain classifiers in the adversarial domain adaptation networks. The structures of the two adversarial domain adaptation networks are shown in Figure 3. Adam optimizer [34] was adopted for training of the networks to obtain faster convergence. We performed randomized search of the hyperparameters over some predefined sets of values. For each method, the hyperparameter settings were evaluated with the leave-one-subject-out cross-validation algorithm and the best setting was chosen to generate the final results. The specific predefined value sets for some of the hyperparameters are listed in Table II.

TABLE II
Value Sets for Hyperparameter Tuning

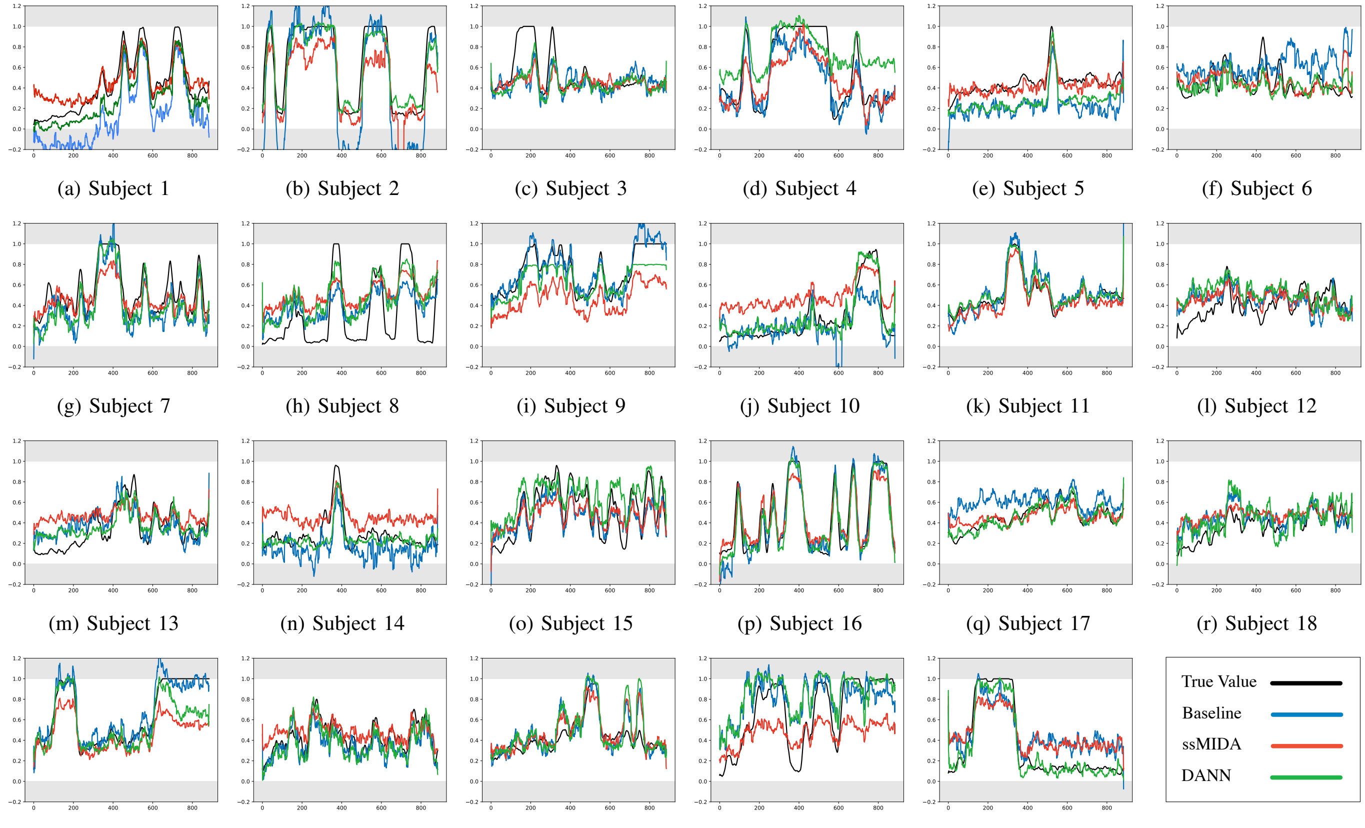
Type	Value Set
Subspace Dimension	$\{10, 20, 40, 60, 80, 100, 120\}$
C for SVR	$\{2^n \mid n \in \{-10, -9, \cdots, 10\}\}$
ϵ for SVR	$\{0, 0.01, \cdots, 0.1\}$
λ for DANN&ADDA	$\{2^n \mid n \in \{-10, -9, \cdots, 10\}\}$
Learning Rate for Adam	$\{2^n \times 10^{-4} \mid n \in \{-10, -9, \cdots, 10\}\}$
Other Hyperparameters	$\{2^n \mid n \in \{-10, -9, \cdots, 10\}\}$

To evaluate the estimation results, *Pearson's correlation coefficient* (PCC) and *root-mean-square error* (RMSE) are used. Their definitions are

$$\mathrm{RMSE} = \sqrt{\frac{1}{n_{\text{target}}} \sum_{i=1}^{n_{\text{target}}} (\hat{y}_i - y_i)^2} \tag{5}$$

TABLE III
RESULTS OF DOMAIN ADAPTATION

		Baseline	GFK	SA	TCA	MIDA	SSTCA	SSMIDA	DANN	ADDA
PCC	AVG	0.7606	0.7907	0.7707	0.7786	0.7858	0.7722	0.8024	0.8402	**0.8442**
	STD	0.2314	0.1260	0.0745	0.2152	0.1900	0.2061	0.1629	0.1535	0.1336
RMSE	AVG	0.1689	0.1910	0.1667	0.1596	0.1840	0.1607	0.1701	0.1427	**0.1405**
	STD	0.0673	0.0636	0.0746	0.0544	0.0753	0.0513	0.0686	0.0588	0.0514



(a) Subject 1 (b) Subject 2 (c) Subject 3 (d) Subject 4 (e) Subject 5 (f) Subject 6

(g) Subject 7 (h) Subject 8 (i) Subject 9 (j) Subject 10 (k) Subject 11 (l) Subject 12

(m) Subject 13 (n) Subject 14 (o) Subject 15 (p) Subject 16 (q) Subject 17 (r) Subject 18

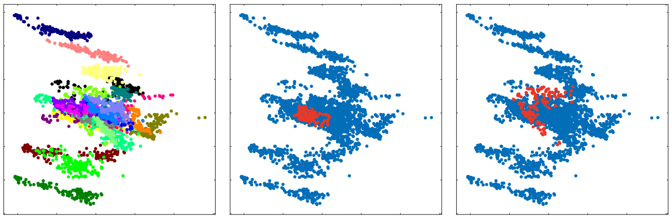
(s) Subject 19 (t) Subject 20 (u) Subject 21 (v) Subject 22 (w) Subject 23 (x) Legend

Fig. 4: Vigilance estimations (in terms of PERCLOS indices) of different subjects using two domain adaptation methods. The x-axes correspond to elapsed times, and the y-axes correspond to the estimated PERCLOS index values. Higher values indicate lower vigilance levels. Each figure corresponds to the estimates for one of the subjects. As is shown in the figures, the black lines are ground truth PERCLOS index values obtained from eye tracking glasses. The blue lines, the red lines, and the green lines are estimates provided by the baseline approach, the SSMIDA method, and the DANN method, respectively.

and

$$\text{PCC} = \frac{\sum_{i=1}^{n_{\text{target}}}(\hat{y}_i - \bar{\hat{y}}_i)(y_i - \bar{y})}{\sigma_{\hat{y}}\sigma_y}, \tag{6}$$

where $\hat{y}_i$ is the predicted value, y_i is the true value, $\sigma_{\hat{y}}$ and σ_y are the corresponding standard deviations. While RMSEs show the average error of the estimates, PCCs are related to structural relationships between the estimates and the labels. Typically, smaller values of RMSEs or bigger values of PCCs indicate better performance.

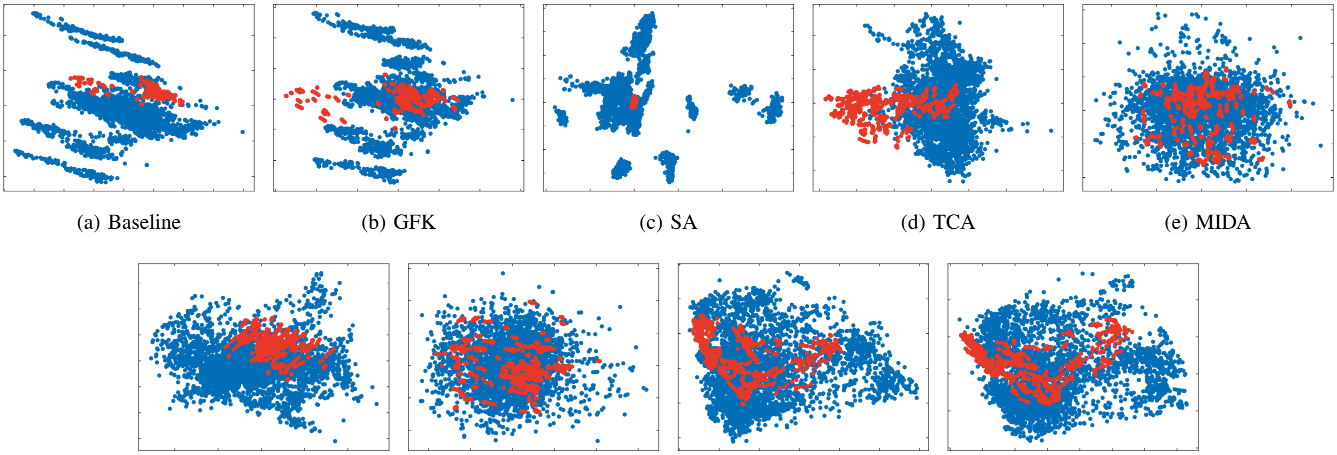


(a) All subjects (b) Subject 3 (c) Subject 8

Fig. 5: Illustrations of original feature distributions.

B. Domain Adaption Results

In Table III, the averages (AVGs) and standard deviations (STDs) of PCCs and RMSEs using different domain adaptation methods are described. The adversarial domain adaptation networks achieve significant improvement in performance both in terms of PCC (0.8402 and 0.8442 compared with baseline's 0.7606, p-values being 0.0121 and 0.0091) and in terms of RMSE (0.1427 and 0.1405 compared with baseline's 0.1689, p-values being 0.0557 and 0.0131), mostly at 0.05 level. For the adversarial domain adaptation networks, ADDA performs slightly better than DANN. Among the traditional methods,



(a) Baseline (b) GFK (c) SA (d) TCA (e) MIDA

(f) SSTCA (g) SSMIDA (h) DANN (i) ADDA

Fig. 6: Plots of distributions of features after domain adaptation. Blue points indicate data points from source domains, while red ones indicate data points from target domains.

SSMIDA outperforms other methods in terms of PCC (0.8024) while TCA performs the best in terms of RMSE (0.1596).

In Figure 4, the vigilance estimation results (i.e., predictions of the PERCLOS index values) of different subjects using two domain adaptation methods are plotted. DANN and SSMIDA are chosen to represent adversarial domain adaptation networks and traditional methods, respectively. Besides, the true labels and the estimates provided by the baseline approach are also plotted for comparison. The figures show that all of the three methods can output estimates that follow the vigilance changing trends, and DANN achieves the best performance under most of the cases.

C. Discussions

By observing the results mentioned above, the following conclusions can be derived. (1) In Table III, though all of the domain adaptation methods achieved better performance than the baseline method in terms of PCC, some of them (MIDA, GFK, SSMIDA) failed to achieve better performance in terms of RMSE. Considering the properties of PCC and RMSE, the three methods can output estimates that follow the vigilance changing trends but with larger errors. (2) In Figure 4, for most subjects, DANN performs better than SSMIDA, and both of them perform better than the baseline method in estimating the true labels. This is consistent with the results shown in Table III. There are cases where the estimates are smaller than 0 or larger than 1. This mostly happens for the baseline estimation. The reason is that the large domain discrepancies were not reduced by any domain adaptation methods. (3) There are a few cases when all of the three methods could not achieve good performance. Two examples are shown in Figures 4(c) and 4(h) where the estimates are inaccurate for some of the large and small PERCLOS index values. This phenomenon is possibly caused by individual differences shown in Figure 5. The domain discrepancies are shown by plotting the feature distributions (before domain adaptation) in a two-dimensional space derived by applying the PCA algorithm. Data points from subjects 3 and 8 are emphasized in Figures 5(b) and 5(c). It can be observed that the distributions of subjects 3 and 8 are very different from those of other subjects. This indicates huge individual differences (domain discrepancies) which should account for the undesirable performance of domain adaptation methods on these two subjects.

To unveil the influence of domain adaptation methods on the feature distributions, the output features of all the domain adaptation methods (with subject 1 set as the target domain and other subjects set as the source domain) are plotted in Figure 6. The two-dimensional spaces are obtained by applying the PCA algorithm. From the figures, following conclusions can be obtained. (1) From Figure 6(a), it can be observed that the original features from different domains are in different distributions. This is the case which was introduced in Section II: $P(X_s) \neq P(X_t)$. (2) After applying most of the domain adaptation methods, the distributions become similar to each other. This means that the domain adaptation objective has been achieved: $P(\phi(X_s)) \approx P(\phi(X_t))$. (3) SA fails to align the distributions into similar ones, which explains the relatively low PCC as shown in Table III. (4) The distributions of output features in Figures 6(e), 6(g), 6(h), and 6(i) are successfully aligned, which is consistent with the good performance of MIDA, SSMIDA, DANN and ADDA as shown in Table III. (5) For the domain adaptation methods that are able to align multiple source and target domains simultaneously (i.e., MIDA, SSMIDA, and DANN), data from all of the 23 subjects are successfully aligned to similar distributions.

V. Conclusions

In this paper, we have introduced adversarial domain adaptation networks for multimodal cross-subject vigilance estimation. The recently proposed domain-adversarial neural network (DANN) and adversarial discriminative domain adaptation (ADDA) were adopted, and both of them have achieved considerable improvement in estimation accuracy in comparison with other existing domain adaptation methods. Experimental results have demonstrated that the domain adaptation methods can reduce domain discrepancies by aligning the distributions of the data from different subjects into similar distributions.

Acknowledgement

This work was supported in part by grants from the National Key Research and Development Program of China (Grant No. 2017YFB1002501), the National Natural Science Foundation of China (Grant No. 61673266), the Major Basic Research Program of Shanghai Science and Technology Committee (Grant No. 15JC1400103), ZBYY-MOE Joint Funding (Grant No. 6141A02022604), the Technology Research and Development Program of China Railway Corporation (Grant No. 2016Z003-B), and the Fundamental Research Funds for the Central Universities.

References

[1] National Center for Statistics and Analysis, "Drowsy driving 2015," *Washington, DC: National Highway Traffic Safety Administration*, Oct 2017.

[2] R. Jenke, A. Peer, and M. Buss, "Feature extraction and selection for emotion recognition from EEG," *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp. 327–339, July 2014.

[3] H. Adeli, S. Ghosh-Dastidar, and N. Dadmehr, "A wavelet-chaos methodology for analysis of EEGs and EEG subbands to detect seizure and epilepsy," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 2, pp. 205–211, Feb 2007.

[4] W. L. Zheng and B. L. Lu, "Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, Sept 2015.

[5] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 42–55, Jan 2012.

[6] S. K. D'mello and J. Kory, "A review and meta-analysis of multimodal affect detection systems," *ACM Computing Surveys*, vol. 47, no. 3, pp. 43:1–43:36, Feb. 2015.

[7] W. Liu, W.-L. Zheng, and B.-L. Lu, "Emotion recognition using multimodal deep learning," in *International Conference on Neural Information Processing*. Springer International Publishing, 2016, pp. 521–529.

[8] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, "Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 121–131, Jan 2011.

[9] C. T. Lin, C. H. Chuang, C. S. Huang, S. F. Tsai, S. W. Lu, Y. H. Chen, and L. W. Ko, "Wireless and wearable EEG system for evaluating driver vigilance," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 2, pp. 165–176, April 2014.

[10] R. Chai, G. R. Naik, T. N. Nguyen, S. H. Ling, Y. Tran, A. Craig, and H. T. Nguyen, "Driver fatigue classification with independent component by entropy rate bound minimization analysis in an EEG-based system," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 715–724, May 2017.

[11] W.-L. Zheng and B.-L. Lu, "A multimodal approach to estimating vigilance using EEG and forehead EOG," *Journal of Neural Engineering*, vol. 14, no. 2, p. 026017, 2017.

[12] X.-Q. Huo, W. L. Zheng, and B. L. Lu, "Driving fatigue detection with fusion of EEG and forehead EOG," in *International Joint Conference on Neural Networks*, July 2016, pp. 897–904.

[13] N. Zhang, W.-L. Zheng, W. Liu, and B.-L. Lu, "Continuous vigilance estimation using LSTM neural networks," in *International Conference on Neural Information Processing*. Springer International Publishing, 2016, pp. 530–537.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, vol. 27, pp. 2672–2680.

[15] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, vol. 37. Proceedings of Machine Learning Research, 07–09 Jul 2015, pp. 1180–1189.

[16] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 2962–2971.

[17] Y.-Q. Zhang, W.-L. Zheng, and B.-L. Lu, "Transfer components between subjects for EEG-based driving fatigue detection," in *International Conference on Neural Information Processing*. Springer International Publishing, 2015, pp. 61–68.

[18] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb 2011.

[19] C. S. Wei, Y. P. Lin, Y. T. Wang, T. P. Jung, N. Bigdely-Shamlo, and C. T. Lin, "Selective transfer learning for EEG-based drowsiness detection," in *IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2015, pp. 3229–3232.

[20] D. Wu, C. H. Chuang, and C. T. Lin, "Online driver's drowsiness estimation using domain adaptation with model fusion," in *International Conference on Affective Computing and Intelligent Interaction*, Sept 2015, pp. 904–910.

[21] D. Wu, V. J. Lawhern, S. Gordon, B. J. Lance, and C. T. Lin, "Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (owarr)," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1522–1535, Dec 2017.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[23] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 2066–2073.

[24] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, Dec 2013, pp. 2960–2967.

[25] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*. MIT Press, 2007, vol. 19, pp. 513–520.

[26] K. Yan, L. Kou, and D. Zhang, "Learning domain-invariant subspace using domain features and independence maximization," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 288–299, Jan 2018.

[27] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *International Conference on Algorithmic Learning Theory*. Springer Berlin Heidelberg, 2005, pp. 63–77.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks." *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[29] Y. F. Zhang, X. Y. Gao, J. Y. Zhu, W. L. Zheng, and B. L. Lu, "A novel approach to driving fatigue detection using forehead EOG," in *International IEEE/EMBS Conference on Neural Engineering*, April 2015, pp. 707–710.

[30] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287 – 314, 1994.

[31] A. Bulling, J. A. Ward, H. Gellersen, and G. Troster, "Eye movement analysis for activity recognition using electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, April 2011.

[32] X. Y. Gao, Y. F. Zhang, W. L. Zheng, and B. L. Lu, "Evaluating driving fatigue detection algorithms using eye tracking glasses," in *International IEEE/EMBS Conference on Neural Engineering*, April 2015, pp. 767–770.

[33] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, Jun. 2008.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computing Research Repository*, vol. abs/1412.6980, 2014.