Integrating and Decomposing Manipulations for Generalized EEG-based Emotion Recognition

Xinyu Xu 518021910645
xuxinyu2000@sjtu.edu.cn
Department of Computer Science and Engineering, Shanghai Jiao Tong University

Abstract—EEG-based affective computing is a core technique in building up an intelligent machine and many studies validate its efficiency and effectiveness in emotion recognition. However, EEG signals vary dramatically among different humans, which makes machine learning model suffer a lot. To tackle this problem, apart from conventional domain adaption methods, we propose a novel domain generalization neural architecture which is agnostic to the target domain distribution. The framework is composed of the **Integrating Decomposing Network (IDN)** and the following **LSTM**. In the IDN module, decomposer splits the EEG signal into distilled emotion representation and domain identification encoding, while integrator guides the self-reconstruction. We use unsupervised pre-training technique to train IDN with the objective to narrow the gap between cross-domain emotion representation as well as the robustness in self-reconstruction. Next, IDN-extracted emotion representation is sent to the following LSTM for prediction, considering both spatial and temporal configuration. We conduct experiments and make detailed analysis on three emotion classification. Our framework works well and outperforms the SVM baseline a lot. Code is available in my submission folder.

Index Terms—Emotion Recognition, EEG (Electroencephalogram), Transfer Learning, Domain Generalization, Integrating and Decomposing, Maximum Mean Discrepancy, LSTM

1 Introduction

Emotion is a basic element in building up real intelligent applications, such as chatting robots. In the early developing stage of artificial intelligence, emerging affective computing paves the way for the robotic emotion simulation in future. Besides, it also helps in detecting and responding to human's mental state, including vigilance estimation and othe brain-computer interfaces. Therefore, it's crucial to build up a general affective computing machine.

Recently, EEG (Electroencephalogram) signals [1] [2], eye movement [3] and EOG (Electrooculogram) signals [4] become mainstream data source in affective computing and attract lots of attention in cutting-edge researches. Also, some studies [5] [6] [7] prove that the multi-modal fusing or combination improves machine cognition and has a larger potential in relevant applications. But in this paper, we only use the single modal EGG signals for emotion recognition.

Many works highlight how to select feature and classifier to predict emotion in spatial-temporal channels but fail to set-up personalized configurations. In fact the EEG signals in different human domains varies a lot and it can't be modeled by simple classification, which also leads into bad performance. It also prevents machine to achieve general affective cognition.

An efficient knowledge transfer from the training human domain to the testing target domain will be a solution to tackle this problem, formally named as transfer learning [8]. There are many applications about transfer learning in many fields, such as cross-category animal pose estimation. As for affective computing, Zheng et.al [9] learns a mapping from domain distribution to a domain-specific classifier. Zhao et.al. [10] proposed a plug-and-play adaption with a short calibration term to enforce the model to adapt the target domain. Besides, Luo et.al [11] use a generative data augmentation method to enrich target domain data by GAN. Generally, these are all domain adaption methods, which requires the overall target domain distribution before knowledge transfer. They are proved to perform well but we may wonder whether there is a more faster method to inference human mental state without any pre-collected information about human characteristic.

In fact, transfer learning can be mainly divided into two parts, domain adaption and domain generalization. Relatively, domain adaption uses the target domain distribution in training while domain generalization not. Both methods have their own advantages but we argue that domain generalization is more appropriate to support wider range of applications, due to its task-agnostic nature.

Therefore, we propose a novel neural architecture to achieve domain generalization with the idea of feature manipulation. We argue that the collected EEG signal is a composition of real emotion representation and domain identification encoding. We build-up an auto-encoder style **Integrating Decomposing Network** (short as **IDN**) to manipulate feature. More detailed, a *decomposer* module decompose EEG feature into two parts, followed by a *integrater* module for self-reconstruction. IDN is pre-trained with unsupervised learning technique intended to minimize the domain gap of extracted real emotion representation as well as a robust reconstruction process. Then, after the pre-training, the distilled emotion representation is sent to a post-**LSTM** with a classifier to get the final prediction according to both the spatial-temporal configuration.

We conduct 15-fold cross-validation experiments on classifying three emotions. Comparing to the SVM baseline, our framework outperforms a lot. We also make detailed analysis to validate its efficency and effectiveness.

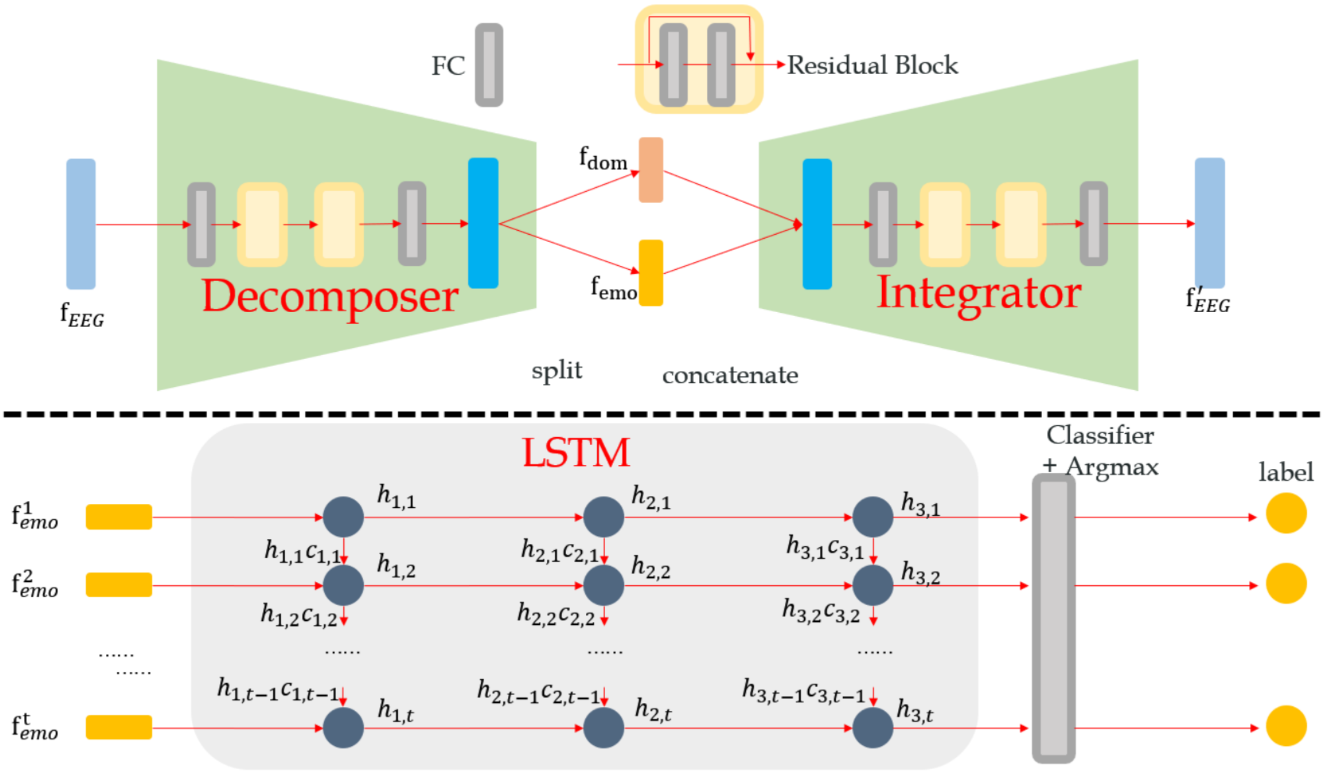

Fig. 1. Overview of our proposed architecture. First IDN is used to extract distilled emotion representation f_{emo}. Inside IDN, the decomposer and integrator are both deep MLP network with residual connection. Then f_{emo} is sent to the following LSTM to give predictions.

2 Related Work

2.1 Emotion Recognition

Emotion recognition attracts many researchers. Zheng et.al. [12] construct EEG-based emotion recognition dataset SEED for three emotions: positive, neutral and negative, from 15 subjects in 2015. And they introduce deep belief networks in this benchmark. Peng et.al. [13] adopt manifold algorithm in emotion classification. Also, some works fuse EEG signals with other modality and achieve better performance on emotion recognition. Zhao et.al use a dense co-attention symmetric network. Lan et.al. [14] propose a novel method about deep generalized canonical correlation analysis.

2.2 Domain Adaption

Given the target domain distribution, a well-trained model on the source domain gets fine-tine again. This is called domain adaption. It can be classified into homogeneous and heterogeneous domain adaption forms. In the homogeneous domain adaption, data space between domains are very similar, domain shift is practically used in this scenario. However, heterogeneous domain adaption is much more complex. As its application in personalizing emotion recognition, Zheng et.al. [9] introduce a mapping from the data distribution into the domain-specific classifier. Zhao et.al. [10] propose a plug-and-play carlibration mechanism to compress the time for adaption. Mu et.al. use the domain adaption technique in personalizing EEG-based sleep quality evaluation.

2.3 Domain Generalization

Domain generalization aims at developing model robust to target domain without prior information about data distribution in target domain. There are three common methods generally, namely feature-based, classifier-based and domain augmentation. The first two methods intend to learn domain invariant feature or train domain invariant classifier while the last one achieve domain-invariant by generating massive pseudo-domains in training. Domain generalization is a much harder problem than domain adaption. But for real world application, it's impossible to know the target domain before using. Therefore, domain generalization is a more potential benchmark.

3 Approach

3.1 Overview

EEG signals vary a lot for different human subjects, also vary a lot when different emotion mentality is triggered. Therefore, we emphasize that the collected EEG emotion signal f_{EEG} is a composition of exact emotion encoding f_{emo} and domain-sepcific embedding f_{dom}. Ideally, this can be mathematically represented as

$$f_{EEG} = \Phi(f_{dom}, f_{emo})$$

where Φ is an mapping function for integrating.

Next, we use mapping Ψ to consider the inverse manner, which is also the role of decomposer

$$f_{dom}, f_{emo} = \Psi(f_{EEG})$$

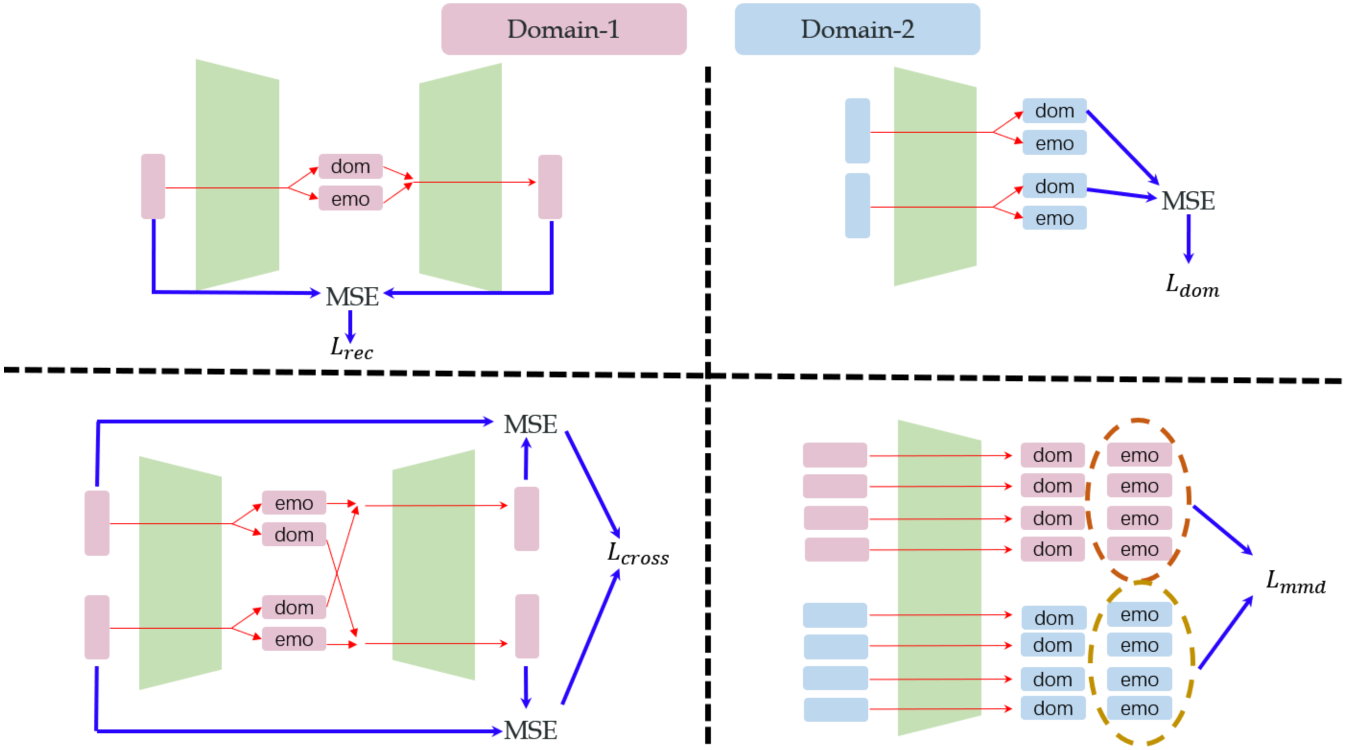

Fig. 2. The unsupervised pre-training phase of Integrating Decomposing Network (IDN). The light red and blue rectangles denote features of two domains. The deep red and blue lines denote the feedforward path and loss computing respectively. We use reconstruction loss L_{rec}, domain consistency loss L_{dom}, cross loss L_{cross} and maximum mean discrepancy loss L_{mmd} as learning objectives. They are free of emotion labels.

With defined feature mapping functions above, we are able to perform feature manipulations. We name the whole module containing decomposer and integrator as **Integrating Decomposing Network (IDN)**.

Most importantly, after IDN is well trained and generalized, we can extract clean emotion representation f_{emo} based on EEG signals f_{EEG}. Next we use a following LSTM to give the final prediction. Apart from direct mapping MLP method, LSTM is able to aggregate emotion in the temporal channel, improving model comprehension in emotion recognition.

The general overview of our model is visualized in Figure 1. We use deep multi layer perceptron with residual connection to implement IDN.

3.2 Unsupervised pre-training of IDN

Here we introduce how to train IDN in an unsupervised manner. The role of IDN is to debias the EEG signal and seek the cleaned emotion feature, not to know what's the exact emotion label, thus it can be free of label.

Reconstruction loss

The IDN is an auto-encoder structured neural network, therefore, we can refer to the self-supervised training phase of auto-encoder. That is, EEG feature should be consistent after the manipulation of decomposer and integrator. The self-reconstruction functionality should be robust.

Therefore, we introduce the first learning objective, reconstruction loss L_{rec} via measure the MSE distance between the EEG input and what reconstructed.

$$L_{rec} = ||f_{EEG} - \Phi(\Psi(f_{EEG}))||_2$$

Domain consistency loss

The decomposer Ψ will transform and split f_{EEG} into the exact emotion encoding f_{emo} as well as the domain-specific embedding f_{dom}. Here we consider f_{dom} is only determined by the domain subject and all instances in the same domain should share the same domain identification encoding.

Therefore, we introduce the second learning objective, named as domain consistency loss L_{dom}, which measures the MSE distance of decomposed domain identification encoding for pair instances in the same domain. Assume f^a_{EEG} and f^b_{EEG} are two EEG signals for one person in two periods. Then we formulate L_{dom} as

$$f^a_{dom}, f^a_{emo} = \Psi(f^a_{EEG})$$
$$f^b_{dom}, f^b_{emo} = \Psi(f^b_{EEG})$$
$$L_{dom} = ||f^a_{dom} - f^b_{dom}||_2$$

Cross loss

Next, we consider both the reconstruction consistency and domain consistency to make a combination. That is, as long as the domain identification encoding of all instances in the same domain are ideally same. Then, if we use randomly cross f_{dom} of each instances, the corresponding reconstruction phase still be consistent.

Similarly, assume f^a_{EEG} and f^b_{EEG} are two EEG signals for one person in two periods. We model the third objective, cross loss as

$$\begin{aligned} f^a_{dom}, f^a_{emo} &= \Psi(f^a_{EEG}) \\ f^b_{dom}, f^b_{emo} &= \Psi(f^b_{EEG}) \\ L_{cross} =& ||f^a_{EEG} - \Phi(f^b_{dom}, f^a_{emo})||_2 \\ &+ ||f^b_{EEG} - \Phi(f^a_{dom}, f^b_{emo})||_2 \end{aligned}$$

Maximum mean discrepancy loss

Last, we introduce a widely used loss function in transfer learning, named as maximum mean discrepancy. It is used with intention to narrow the gap between domains via benchmarking the statistical maximum mean discrepancy. We are going to introduce detailed computation about it next.

Initially, the maximum mean discrepancy is defined as

$$MMD(p, q, \mathcal{H}) = \sup_{||f||_{\mathcal{H}} \leq 1} E_p(f(x)) - E_q(f(y))$$

where p, q are the data distribution of two domains, x, y are data in two domains respecting to the distribution p, q respectively. $\mathcal{H}$ is the Reproducing Kernel Hilbert Space (RKHS) and function f is the kernel function. We can find that MMD measures difference of two data distribution, it decreases to zero if and only if two distributions are totally same.

Then we transform the representation of expectation as

$$\begin{aligned} E_p(f(x)) &= \int_{\mathcal{X}} p(dx) f(x) \\ &= \int_{\mathcal{X}} p(dx) < k(x, \cdot), f >_{\mathcal{H}_k} \\ &=< \int_{\mathcal{X}} p(dx) k(x, \cdot), f >_{\mathcal{H}_k} \\ &=< \mu_p, f >_{\mathcal{H}_k} \end{aligned}$$

First, the expectation can be written in the form of integral. Second, referring to the kernel trick, $f(x)$ can be reformed as the inner product of the extended vector $k(x, \cdot)$ with the base vector f. Here $< \cdot, \cdot >$ is the inner product. Third, we take inside the integral according to the commutative law. Last, it can be seen as the inner product of the kernel space data distribution expectation μ_p with base vector f.

Therefore, MMD can be expressed as

$$\begin{aligned} MMD(p, q, \mathcal{H}) &= \sup_{||f||_{\mathcal{H}} \leq 1} E_p(f(x)) - E_q(f(y)) \\ &= \sup_{||f||_{\mathcal{H}} \leq 1} < \mu_p, f >_{\mathcal{H}_k} - < \mu_p, f >_{\mathcal{H}_k} \\ &= \sup_{||f||_{\mathcal{H}} \leq 1} < \mu_p - \mu_q, f >_{\mathcal{H}_k} \\ &\leq \sup_{||f||_{\mathcal{H}} \leq 1} ||\mu_p - \mu_q||_{\mathcal{H}} \cdot ||f||_{\mathcal{H}} \\ &\leq ||\mu_p - \mu_q||_{\mathcal{H}} \end{aligned}$$

In the last two lines, we consider the inequality of inner product and the equality can be achieved.

In fact, we can't directly get the expectation of ground truth data distribution, thus we replace it with observed data points. Assume $X \sim p, Y \sim q$, there are n instances in X and m instances in q.

$$MMD(X, Y) = ||\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \frac{1}{m}\sum_{i=1}^{m} f(y_i)||_{\mathcal{H}}$$

We can't explicitly compute the RKHS vector, but kernel tricks allow us to obtain implicit distances.

$$\begin{aligned} &MMD(X, Y)^2 \\ =& ||\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \frac{1}{m}\sum_{i=1}^{m} f(y_i)||^2_{\mathcal{H}} \\ =& ||\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} f(x_i)f(x_j) - \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} f(x_i)f(y_j) \\ &+ \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} f(y_i)f(y_j)|| \\ =& ||\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k(x_i, x_j) - \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i, y_j) \\ &+ \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} k(y_i, y_j)|| \end{aligned}$$

This is the final form of MMD which can be directly computed.

In the detailed implementation, we use RBF kernel. To achieve better generalization the capability, we use 5-kernel architecture. That is we use different parameter σ in RBF kernel as [0.25, 0.5, 1, 2, 4], and sum results of each.

Return back to our network, let f^S_{EEG} denote EEG signals in domain S and f^T_{EEG} denote EEG signals in domain T. As our intented, we want the decomposed real emotion feature f_{emo} should be generalized to all domains, thus we use MMD for constraint. It can be formulated as

$$\begin{aligned} f^S_{dom}, f^S_{emo} &= \Psi(f^S_{EEG}) \\ f^T_{dom}, f^T_{emo} &= \Psi(f^T_{EEG}) \\ L_{mmd} &= MMD(f^S_{emo}, f^T_{emo}) \end{aligned}$$

Total loss

The total learning objective for IDN pre-training is a weighed summation of all losses mentioned above. Since different losses have different scales and gradients, we assign different weights to them for stable training. We have

$$L_{pretrain} = \lambda_1 L_{rec} + \lambda_2 L_{dom} + \lambda_3 L_{cross} + \lambda_4 L_{mmd}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are hyper-parameters for tuning.

3.3 LSTM classification

Up to now, we can seek debaised emotion representation f_{emo} which is generalized to all domains. Next we are going to do emotion classification.

We can find the temporal channel concept is important in recognize emotion, since our EEG signals is always continuous as well as the human mental state. If our model is able to capture the emotion property in the temporal channel, it will have great performance boost.

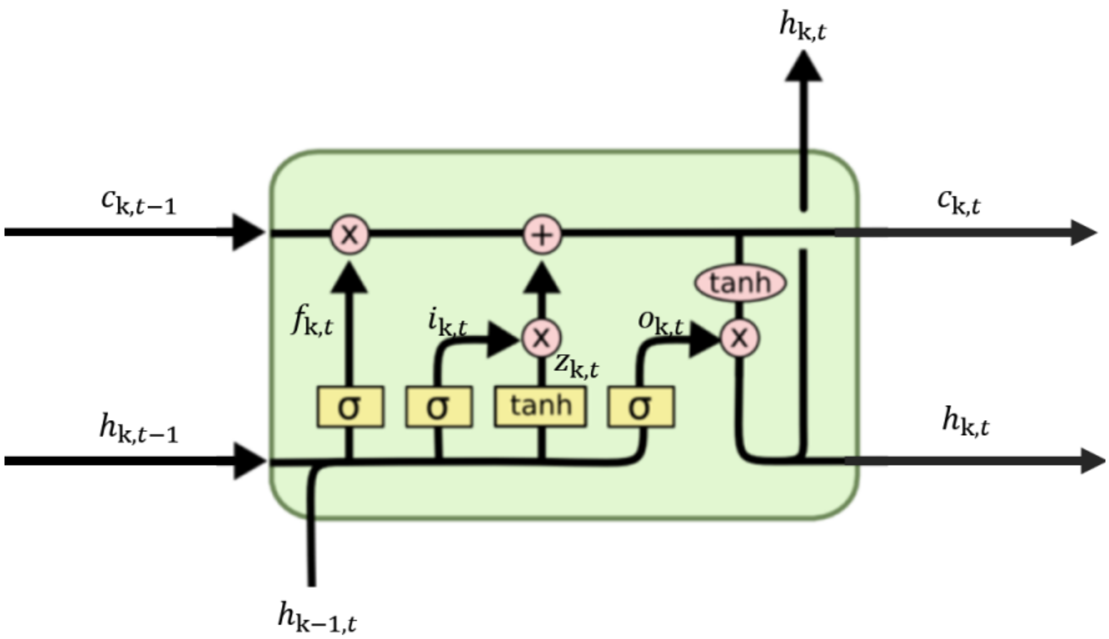

Fig. 3. Illustration of LSTM unit at k-th layer and time t. It takes information from the previous layer $h_{k-1,t}$ and information from last time period $h_{k,t-1}, c_{k,t-1}$ to output the new cell state $c_{k,t}$ with new hidden state $h_{k,t}$.

To model temporal sequence, there are some widely used tools. RNN and LSTM are most widely used and support many down-stream Tasks. Transformer and Informer are recent new neural architecture to model end-to-end sequence, but they are much expensive in computation, which I am not able to afford. Comparing RNN and LSTM, RNN suffers from the gradient vanish and is not appropriate for very long sequence. Finally, We choose LSTM.

A simple illustration of LSTM unit is in Figure 3. We use $h_{k,t}$ to denote the hidden state of k-th layer at time t, and $c_{k,t}$ denotes the cell state of k-th layer at time t. The LSTM unit takes $h_{k-1,t}$, $h_{k,t-1}$ and $c_{k,t-1}$ as input then output $h_{k,t}$ and $c_{k,t}$.

First, LSTM determines what to forget from previous time slice. A linear layer with sigmoid converts $h_{k-1,t}$ with $h_{k,t-1}$ into continuous number in range (0,1). Then it makes element-wise multiplication with $c_{k,t-1}$ to suppress the cell state from the last time. It can be formulated as

$$f_{k,t} = \sigma(W_f[h_{k,t-1}, h_{k-1,t}] + b_f)$$
$$c'_{k,t} = c_{k,t-1} \cdot f_{k,t}$$

where σ is the sigmoid function.

Second, LSTM records and memorizes new information at this time. A linear layer with $\tanh$ converts $h_{k-1,t}$ with $h_{k,t-1}$ into new memory $z_{k,t}$. Another linear layer with sigmoid converts $h_{k-1,t}$ with $h_{k,t-1}$ into the memory selector $i_{k,t}$, whose range in (0,1). $z_{k,t}$ gets element-wise multiplication with $i_{k,t}$ and the result is added to the cell stated which is partially forgotten before. It can be formulated as

$$\begin{aligned} z_{k,t} &= \tanh(W_z[h_{k,t-1}, h_{k-1,t}] + b_z) \\ i_{k,t} &= \sigma(W_i[h_{k,t-1}, h_{k-1,t}] + b_i) \\ c_{k,t} &= c'_{k,t} + z_{k,t} \cdot i_{k,t} \\ &= c_{k,t-1} \cdot f_{k,t} + z_{k,t} \cdot i_{k,t} \end{aligned}$$

Last, the LSTM unit will give the output and update the hidden state. A linear layer with sigmoid converts $h_{k-1,t}$ with $h_{k,t-1}$ into $o_{k,t}$. Next, it gets element-wise multiplication with the tanh activated cell state $c_{k,t}$. It can be formulated as

$$o_{k,t} = \sigma(W_z[h_{k,t-1}, h_{k-1,t}] + b_z)$$
$$h_{k,t} = o_{k,t-1} \cdot \tanh(c_{k,t})$$

In the detailed implementation of the whole model, the depth of LSTM is 3 and hidden dimension is 256.3

The output feature is sent to a linear classifier to give the prediction on three emotions. The LSTM-based classifier is supervised by CrossEntropyLoss L_{cls}. It can be formulated as

$$L_{cls} = \sum_{i=1}^{3} -\mathbf{1}[y = i] \log p_{i|x} - \mathbf{1}[y \neq i] \log(1 - p_{i|x})$$

where x is the EEG signal, $p_{i|x}$ is the probability for it is predicted in class i, y is the target label, $\mathbf{1}[\cdot]$ returns 1 if the statement inside is correct otherwise 0.

4 Experiment

4.1 Dataset and Metric

We conduct experiments on the given dataset, which is a small subset of SJTU Emotion EEG Dataset (SEED). Emotions are triggered when participants are required to watch movies of different keynotes. There are three kinds of emotions overall, positive, negative and neutral.

There are 15 human subjects participating in the data collection phase. All of their EEG signals are at 3394 different time steps. EEG feature are differential entropy feature and the finally collected feature is 310-dimensional.

We adopt accuracy as the evaluation protocol. It is measured by how many feature is classified correctly with the ground truth label.

	Fold-0	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Fold-6	Fold-7	Fold-8	Fold-9	Fold-10	Fold-11	Fold-12	Fold-13	Fold-14	Avg	Std
SVM-Linear	0.5860	0.5592	0.5448	0.7637	0.5386	0.6022	0.5737	0.5250	0.3677	0.4646	0.7298	0.5713	0.8182	0.6452	0.7890	0.6053	0.1202
SVM-RBF	0.6096	0.4941	0.4122	0.6014	0.4281	0.5621	0.5969	0.5675	0.5504	0.6123	0.5006	0.6014	0.9127	0.5524	0.3456	0.5565	0.1232
IDN+LSTM	**0.9947**	**0.9705**	**0.9617**	**0.9926**	**0.9523**	**0.9694**	**0.9870**	**0.9464**	**0.9870**	**0.9915**	**0.9785**	**0.9935**	**0.9906**	**0.9900**	**0.9806**	**0.9791**	**0.0151**

TABLE 1
Results of emotion classification, including the accuracy on each fold, the average and standard deviation.

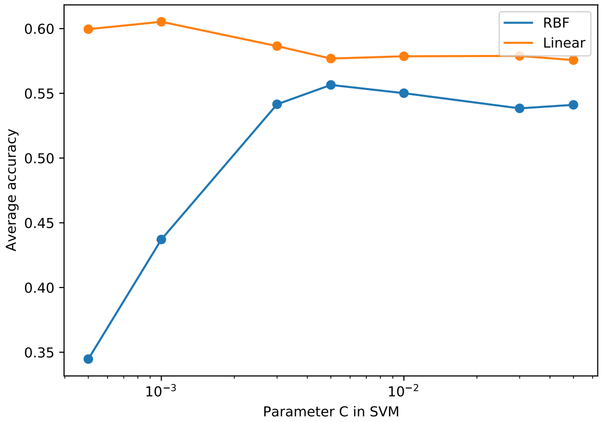

Fig. 4. Results of average SVM classification with different settings of C. Note that X-axis is scaled by log value. Linear kernel SVM is more robust but RBF kernel has worse performance and it drops greatly with small C.

Besides, we conduct on 15-fold cross-validation. That is, in each run, we select one subject for testing while all other subjects are used for training. We repeatedly run experiments with different train-test splits for 15 times. This helps to validate the robustness of a model.

4.2 SVM baseline

To validate the effectiveness and efficiency of proposed method, we also set-up an SVM baseline for comparison. SVM classifies data points via an explicit hyper-plane in the latent space and the model it trained to maximum the margin between data samples and plane. SVM has various kernels, including linear kernel and rbf kernel (radial basis function). Different kernel learns different property of data distribution and their performance varies a lot. Besides, SVM also has different decision making procedure in multi-classification. It can use one-vs-one or one-vs-all strategies in task decomposition. However, we find that the two procedure has similar results, thus we only report what in the one-vs-one setting.

4.3 Implementation details

The main model (IDN+LSTM) is implemented in PyTorch, and trained by a single NVIDIA 1080-Ti GPU. The dimensionality of f_{EEG} and reconstruction f'_{EEG} is 310, while the dimensionality of all all feature (including f_{emo}, f_{dom} and what hides in decomposer, integrator and LSTM) is 256. The decomposer and integrator are both consists of one FC-ReLU layer, two residual blocks and one FC-ReLU layer in order. There are 3 hidden layers in LSTM.

In the first pre-training stage, IDN is supervised by L_{total}. We assign different loss weights for each items, detailed as $\lambda_1 = 1e-3$ (for L_{rec}), $\lambda_2 = 1e-2$ (for L_{dom}), $\lambda_3 = 1e-2$ (for L_{cross}), $\lambda_4 = 1$ (for L_{mmd}). We use Adam optimizer to optimize the model with the learning rate 1e-3. We use 100 epochs for pre-training.

In the second stage, we train the LSTM-based classification and fine-tune the IDN. Here we directly use the classification loss for supervision. We use Adam optimizer to optimize the model with the learning rate 1e-3. We set up the maximum epoch as 300. However, we notice the jitter when model is close to optimal solution, this suggests the overfitting of model and hurts performance. Thus, we use an early stopping criterion that enforce to stop the training when L_{cls} of the training set is less than 1e-3.

4.4 Result

The average SVM classification results of both linear and rbf kernels are reported in Fig 4. We can find that linear kernel is generally better than the rbf kernel, which suggests the linear nature of exact data distribution. From the plot of rbf SVM, we learned that hyper-parameter *C* is important in the classification, which greatly affects performance. An appropriate trade off between loss terms is crucial in building up high-performance model.

We also report the performance of IDN+LSTM, comparing with best SVM baseline in Tab 1. Here we not only report accuracy on 15 folds, but also the average and standard deviation. Our proposed main method almost makes perfect prediction, with 95+% accuracy generally, which proves the effectiveness and efficiency of our model. Comparing with SVM baseline, IDN+LSTM outperforms a lot. This is not only because their model sizes inevitably vary a lot, but also due to the generalized nature of IDN+LSTM. It validates that transfer learning or more specifically domain

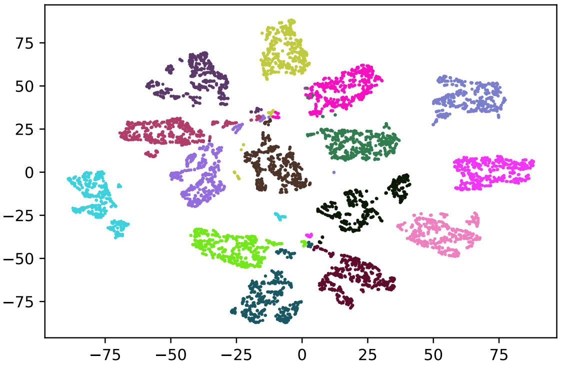

Fig. 6. T-SNE visualization of domain identification encoding f_{dom}. Different colors denote embedding of different domains.

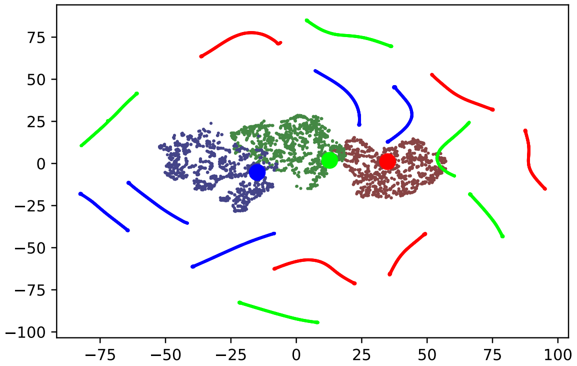

Fig. 7. T-SNE visualization of EEG feature f_{EEG} and the reconstruction f'_{EEG} on three subjective. Light color is initial signal, dark color is the corresponding reconstruction. The central dot is the mean of initial signal.

generalization is an effective technique in handling cross-domain tasks.

4.5 Visualization

To validate how IDN pre-training procedure works, we use t-SNE technique to visualize the feature in the latent space in Figure 6 and Figure 7.

First, Figure 6 is about the domain identification encoding f_{dom}. For f_{dom} of all instances in all domains, we use t-SNE to map the feature into 2-D space and scatter them in a plane. Dots in the same color are instances in the same domain. We can find that instances are appropriately clustered regarding to the domain identification. This validates how the decomposing network recognizes the domain identification.

Next, we visualize the relation between EEG signals and their reconstructions via IDN in Figure 7. Here light red, green, blue plots are initial EEG signals, and dark clusters in the center are their corresponding reconstructions. We find that the original signals are discrete for the same subjective, but the decomposing and integrating manipulations map them into a continuous space. Since the original signals float in a wide range, it's hard for model to learn strictly invariant reconstruction. But if we compute the centroid of the initial signals for each domain, as the large dot in the center of Figure 7, we find it the clustered reconstruction is close to it. This validates decomposing and integrating manipulations reconstruct feature at a certain level.

5 Discussion

In this paper, we propose a new neural architecture IDN+LSTM for emotion recognition, where the auto-encoder style unsupervised pre-trained IDN is used to narrow the gap of multi-domain emotion representation and LSTM models the temporal configuration. We conduct 15 fold cross-validation experiments on the given small dataset. Our method works well and outperforms SVM baseline in a large margin. This validates the effectiveness of our method.

Apart from the high performance, we argue some major advantages of our model next. **(1)** Our method is to achieve target-agnostic domain generalization, which not requires about the target domain data distribution. Comparing with other domain adaption methods, this can support more real-world applications. **(2)** The learning procedure of IDN is unsupervised, both the decomposer and integrator. This means that with out any emotion labels, we can directly obtain the distilled emotion representation as well as the domain identification encoding in a unsupervised manner. **(3)** The EEG signals is a sequence. We notice the continuous nature of EEG signal and use LSTM to model temporal information.

Next, we are going to discuss about experiments about this task. In fact, it is relatively an easy task, comparing the complete emotion classification benchmark. In our first experiment(no reported in the main paper), a deep residual connected MLP model with Batchnorm and ReLU can get perfect accuracy. This means that the complex nature of deep networks certainly learns cross-domain relations.

Acknowledgments

The author would like to thank **Prof. Bao-Liang Lu** for his wonderful lecture in the course, and **TA Li-Ming Zhao** for his patient tutor.

References

[1] W. Zheng and B. 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, 2015.

[2] W. Mu and B. L. Lu, "Examining four experimental paradigms for eeg-based sleep quality evaluation with domain adaptation," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 5913–5916.

[3] Y. Jiao, Y. Peng, B. Lu, X. Chen, S. Chen, and C. Wang, "Recognizing slow eye movement for driver fatigue detection with machine learning approach," in *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*. IEEE, 2014, pp. 4035–4041.

[4] X. Zhu, W. Zheng, B. Lu, X. Chen, S. Chen, and C. Wang, "Eog-based drowsiness detection using convolutional neural networks," in *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*. IEEE, 2014, pp. 128–134.

[5] J. J. Guo, R. Zhou, L. M. Zhao, and B. L. Lu, "Multimodal emotion recognition from eye image, eye movement and eeg using deep neural networks," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 3071–3074.

[6] Y. T. Lan, W. Liu, and B. L. Lu, "Multimodal emotion recognition using deep generalized canonical correlation analysis with an attention mechanism," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–6.

[7] H. Jiang, X. Guan, W. Zhao, L. Zhao, and B. Lu, "Generating multimodal features for emotion classification from eye movement signals," *Aust. J. Intell. Inf. Process. Syst.*, vol. 15, no. 3, pp. 59–66, 2019.

[8] D. Wu, Y. Xu, and B.-L. Lu, "Transfer learning for eeg-based brain-computer interfaces: A review of progress made since 2016," 2020.

[9] W. Zheng and B. Lu, "Personalizing eeg-based affective models with transfer learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. Kambhampati, Ed. IJCAI/AAAI Press, 2016, pp. 2732–2739.

[10] L. Zhao, X. Yan, and B. Lu, "Plug-and-play domain adaptation for cross-subject eeg-based emotion recognition," in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*. IJCAI/AAAI Press, 2021.

[11] Y. Luo, L. Zhu, Z. Wan, and B. Lu, "Data augmentation for enhancing eeg-based emotion recognition with deep generative models," *CoRR*, vol. abs/2006.05331, 2020. [Online]. Available: https://arxiv.org/abs/2006.05331

[12] W. Zheng and B. 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, 2015.

[13] Y. Peng and B.-L. Lu, "Discriminative manifold extreme learning machine and applications to image and eeg signal classification," *Neurocomputing*, vol. 174, pp. 265–277, 2016.

[14] Y.-T. Lan, W. Liu, and B.-L. Lu, "Multimodal emotion recognition using deep generalized canonical correlation analysis with an attention mechanism," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–6.