Generalizable Sleep Staging via Multi-Level Domain Alignment

Jiquan Wang[1,2], **Sha Zhao**[1,2*], **Haiteng Jiang**[3,4,1], **Shijian Li**[1,2], **Tao Li**[3,4,1], **Gang Pan**[1,2,4*]

[1]State Key Laboratory of Brain-machine Intelligence, Zhejiang University, Hangzhou, China
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3]Department of Neurobiology, Affiliated Mental Health Center & Hangzhou Seventh People's Hospital, Zhejiang University School of Medicine, Hangzhou, China
[4]MOE Frontier Science Center for Brain Science and Brain-machine Integration, Zhejiang University, Hangzhou, China
{wangjiquan, szhao, h.jiang, shijianli, litaozjusc, gpan}@zju.edu.cn

Abstract

Automatic sleep staging is essential for sleep assessment and disorder diagnosis. Most existing methods depend on one specific dataset and are limited to be generalized to other unseen datasets, for which the training data and testing data are from the same dataset. In this paper, we introduce domain generalization into automatic sleep staging and propose the task of generalizable sleep staging which aims to improve the model generalization ability to unseen datasets. Inspired by existing domain generalization methods, we adopt the feature alignment idea and propose a framework called SleepDG to solve it. Considering both of local salient features and sequential features are important for sleep staging, we propose a Multi-level Feature Alignment combining epoch-level and sequence-level feature alignment to learn domain-invariant feature representations. Specifically, we design an Epoch-level Feature Alignment to align the feature distribution of each single sleep epoch among different domains, and a Sequence-level Feature Alignment to minimize the discrepancy of sequential features among different domains. SleepDG is validated on five public datasets, achieving the state-of-the-art performance.

Introduction

Sleep plays an important role in human health. Sleep staging refers to the classification of sleep into different sleep stages, which is crucial to identifying sleep problems and other disorders in humans (**?**). Clinically, sleep stages are scored by doctors or experts using electrical activity recorded from sensors attached to different parts of the body. A set of signals from these sensors is called a polysomnogram (PSG), consisting of multiple physiological signals recorded, such as electroencephalogram (EEG) and electrooculogram (EOG). According to the American Academy of Sleep Medicine (AASM) sleep standard (**?**), PSG is usually segmented into 30-second epochs, which are manually classified into five different sleep stages (Wake, N1, N2, N3, and REM) by experts.

In recent years, with the development of deep learning techniques (**????**), many deep learning models (**?**) have been proposed to solve the task of automatic sleep staging. They are implemented with different network structures based on sequence-to-sequence framework, obtaining good performance in sleep staging from PSG recordings. However, most of existing methods adopt the intra-dataset scheme, where the training data and testing data are from the same dataset, ignoring the discrepancies between various datasets and making it difficult to be generalized to unseen sleep staging datasets well (**?**). The discrepancies among multiple datasets could be caused by many factors, such as heterogeneous patient population, different signal channel, different data collection equipment types or manners, or different medical environments. In clinic, a sleep staging model should better be generalized to unseen datasets of any new populations in any environment. Some methods (**???**) illustrate that the training and testing samples are not drawn from the same probability distribution which cause **performance deterioration** when directly applying models trained on source datasets to target datasets. They adopt the ideas of Domain Adaptation (DA) to solve it, but the generalization problem still exist because DA methods assume that the target sleep staging database is accessible.

In order to solve the generalizing-to-unseen-database problem of automatic sleep staging, we introduce the idea of **domain generalization (DG)** into the sleep staging task: A *domain* is composed of data that is sampled from a join distribution of input space and label space. The goal of domain generalization is to learn a model from one or several different but related domains (i.e., diverse training datasets) that will perform well on unseen testing domains (**?**). The unseen domain is *inaccessible* in training procedure. Inspired by some studies (**??**) exploring hospital-level DG in medical imaging, we set a single sleep staging dataset (usually from one hospital) as a *domain* and set the discrepancies between different datasets as *domain shift*. Then multiple datasets used for training can be regarded as *source domains* and the dataset used for testing can be seen as *target domain* or *unseen domain*. Our goal is to learn a model from several source domains (several sleep datasets) that can perform well at the unseen domain (unseen sleep dataset). We name this task as **generalizable sleep staging**.

There have been some existing DG studies (**?????**) in other applications, which mainly focus on learning domain-invariant feature representations by sample-level feature distribution alignment to solve the sample-level classification

*Corresponding authors.

problem, referring to mapping single sample into the corresponding label. However, different from sample-level classification problem, sleep staging is a sequence-to-sequence classification problem, which maps a sequence of samples into the corresponding sequence of labels. According to AASM standard (**?**), not only local salient waveforms within each epoch but also transition patterns of sleep stages between neighbor epochs play critical roles in sleep staging. Meantime, there have been some studies (**??**) proving the importance of both local intra-epoch features and global inter-epoch features for sleep staging. Therefore, it is critical to combine intra-epoch feature (epoch-level) alignment and inter-epoch feature (sequence-level) alignment in alignment process, to solve the generalization problem of sleep staging.

In this paper, we introduce DG to sleep staging and propose a framework, SleepDG, to solve the generalization problem of sleep staging. Our contributions are as follows:

- We introduce a novel task of generalizable sleep staging to solve the generalization problem of sleep staging in clinic. For the task, we propose **SleepDG, a DG-based deep learning framework,** which learns domain-invariant feature representations from different PSG datasets and can perform well on unseen datasets.
- We propose a **Multi-level Feature Alignment** method consisting of **Epoch-level Feature Alignment** and **Sequence-level Feature Alignment** designed to align feature distribution within each single epoch and between epochs in a sleep sequence among different PSG datasets.
- SleepDG is validated on **five public datasets** in the DG scenarios, achieving the state-of-the-art performance.

Related Work

Automatic Sleep Staging

Automatic sleep staging (**?**) refers to the classification of sleep epochs into different sleep stages in a sequence-to-sequence fashion. Existing deep learning methods almost use a local extractor for epoch features and a global extractor for sequential context features. For example, **??** utilized CNN to extract local features and Bi-LSTM to encode temporal information. **???** utilized CNN to capture intra-epoch features and the multi-head self-attention to model global temporal context. **?** utilized a fully-CNN Encoder-Decoder architecture to model local salient wave characteristic and sleep transitional rules. **?** proposed a hierarchical RNN named SeqSleepNet to extract local and global features from time-frequency images of multimodal PSG data. **?** proposed SleepTransformer which encodes both local features and temporal features via fully Transformers. **?** proposed SalientSleepNet, which uses a fully CNN based on the U^2-Net to detect the salient waves from multimodal PSG signals and a Multi-scale Extraction Module to capture sleep transition rules. Above methods achieve a high performance on sleep staging on specific datasets. However, it is difficult for them to be generalized to other unseen datasets.

Domain Generalization

DG (**???**) considers the scenarios where there are domain shift among different domains, which is different from domain adaptation (DA) (**??**) because target domain in DG is inaccessible. The DG problem was formally introduced by **??** as a machine learning problem. Learning domain-invariant feature representations by minimizing the feature distribution divergence is a common and effective method for domain generalization. Some studies minimized the feature distribution divergence by minimizing the maximum mean discrepancy (MMD) (**???**), second order correlation (**??**), Wasserstein distance (**?**) and Similarity Metric (**?**) in deep neural network. Some studies minimized the feature distribution divergence by adopting Domain-adversarial neural network (DANN) (**?**) for learning domain-invariant features, which was originally proposed by **?** for DA. Most of above methods learn domain-invariant features by minimizing single-sample features divergence, neglecting the crucial role of inter-epoch features in sleep staging, which needs to be improved.

Problem Formulation

We introduce a novel task, called generalizable sleep staging, which is a multi-source DG problem. Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the target space, a *domain* is defined as data sampled from a join distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$. Here, **we treat one single sleep dataset as one domain**. We have several labeled source datasets $D^{S_i} = (X^{S_i}, Y^{S_i})$, where $i \in [1, 2, 3, ..., M]$ denotes the i-th source domain and $M = 4$ in this work. $\mathbf{x}_j^{S_i} = \{x_{j,1}^{S_i}, x_{j,2}^{S_i}, ..., x_{j,L}^{S_i}\}$ is the j-th sequence composed of L sleep epochs and $\mathbf{y}_j^{S_i} = \{y_{j,1}^{S_i}, y_{j,2}^{S_i}, ..., y_{j,L}^{S_i}\}$ is the j-th sequence composed of corresponding L sleep stages. $x_{j,k}^{S_i} \in \mathbb{R}^{n \times C}$, n denotes the number of sampling points in a epoch and C is the number of channels. $y_{j,k}^{S_i} \in \{0, 1\}^N$ and $N = 5$ denote the number of *sleep stages (Wake, N1, N2, N3, REM)*. We also have an unseen target dataset $D^T = (X^T, Y^T)$. In the domain shift scenarios, we supposed the different source datasets (X^{S_i}, Y^{S_i}) and target dataset (X^T, Y^T) are sampled from different distributions.

Our generalizable sleep staging task is defined as learning a mapping function $h : \mathcal{X} \rightarrow \mathcal{Y}$ using only source datasets so that the predictive error on an unseen target dataset is minimized. We decompose the prediction function h as $h = f \circ g$, where g is a feature encoder and f is a classifier. Therefore, the goal of minimizing the predictive error on an unseen datasets domain can be formulated as:

$$\min_{f,g} \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim P(X^T, Y^T)} \mathcal{L}(f(g(\mathbf{x})), \mathbf{y}) + \lambda \mathcal{L}_{\text{reg}} \tag{1}$$

where $\mathcal{L}_{\text{reg}}$ denotes some regularization term and λ is the tradeoff parameter.

Method

Overview

Our generalizable sleep staging framework SleepDG is summarized in Fig. 1. In the training phase shown in Fig. 1(a),

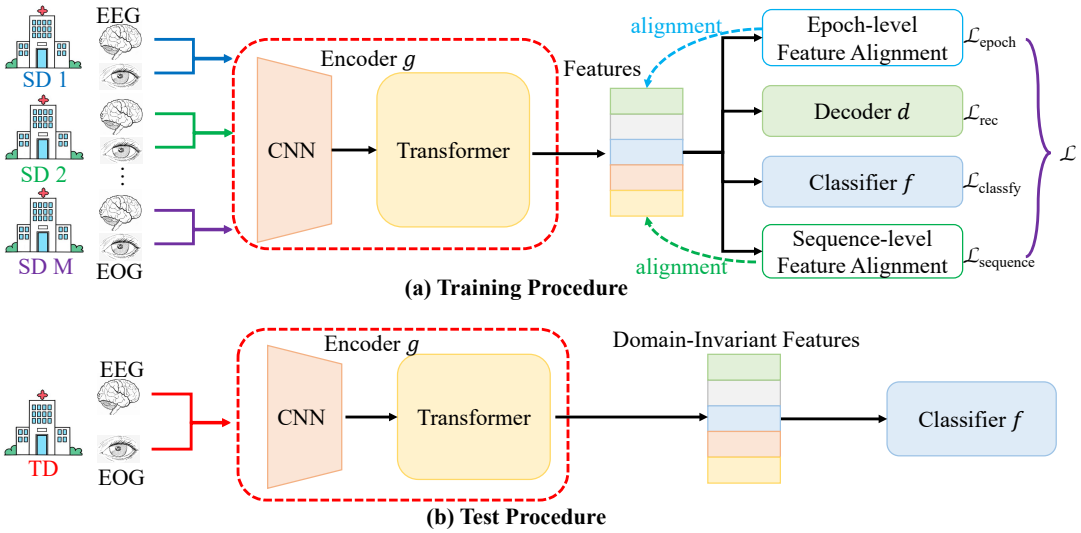


Figure 1: SleepDG overview. Here, SD is source domain and TD is target domain.

we take the source data X^{S_i} which is from M different domain as inputs and use encoder g, decoder d, classifier f and Multi-level Feature Alignment to learn domain-invariant feature representations H^{S_i}. The whole training process is trained in an end-to-end fashion with the supervision of the groundtruth sleep stages. As shown in Fig. 1(b), in the test phase we take the target data X^T which is from the target domain as input and extract the feature representations H^T by the feature encoder g. Then we feed H^T into the classifier f to predict the sleep stages Y^T.

AE-based Feature Encoding

In order to learn feature representations H^{S_i}, we design a sequence-to-sequence autoencoder (AE), consisting of encoder g and decoder d. We take the source data X^{S_i} as inputs and extract the feature representations H^{S_i} by encoder g. The process of features extracting can be expressed as $H^{S_i} = g(X^{S_i})$, where $H^{S_i} = \{\mathbf{h}_j^{S_i}\}_{j=1}^{N^{S_i}}$ is composed of many *sequences* $\mathbf{h}_j^{S_i}$, $\mathbf{h}_j^{S_i} = \{h_{j,1}^{S_i}, h_{j,2}^{S_i}, ..., h_{j,L}^{S_i}\}$ is composed of L feature representations which are learned from L sleep epochs, $h_{j,k}^{S_i} \in \mathbb{R}^d$ and d denotes the feature dimension. Specifically, the feature encoder g consists of CNN and Transformer, where CNN is used to extract intra-epoch features and Transformer is used to extract inter-epoch features. We use a decoder d to reconstruct X^{S_i} from H^{S_i}. The process of reconstruction can be expressed as $\hat{X}^{S_i} = d(H^{S_i})$, where $\hat{X}^{S_i}$ is the reconstructed sleep data corresponding to X^{S_i}. Then, we use the Mean Squared Error (MSE) loss function as the reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \frac{1}{L}\sum_{k=1}^{L} \|x_{j,k}^{S_i} - \hat{x}_{j,k}^{S_i}\|_2, \tag{2}$$

where $x_{j,k}^{S_i}$ is a sleep epoch, $\hat{x}_{j,k}^{S_i}$ is the reconstructed sleep epoch and $\|\cdot\|_2$ denotes the squared norm.

Multi-level Feature Alignment

The feature alignment refers to mapping source data from different datasets into one shared feature space in which $\mathrm{P}(H^{S_i}, Y^{S_i}) = \mathrm{P}(H^{S_j}, Y^{S_j})$ if $i \neq j$. We can decompose joint distribution $\mathrm{P}(H^{S_i}, Y^{S_i})$ into a marginal distribution and a conditional distribution by $\mathrm{P}(H^{S_i}, Y^{S_i}) = \mathrm{P}(H^{S_i})\mathrm{P}(Y^{S_i}|H^{S_i})$. According to AASM standard (**?**), sleep experts adopt a uniform rules for manual sleep staging, so we assume that the conditional distributions are the same in different sleep staging datasets. In other words, we make a prior assumption that $\mathrm{P}(Y^{S_i}|H^{S_i}) = \mathrm{P}(Y^{S_j}|H^{S_j})$, if $i \neq j$. Under such a prior assumption, we only need to align the marginal distribution $\mathrm{P}(H^{S_i})$.

Sleep staging is a sequence-to-sequence classification and both local intra-epoch features and global context features among sequential epochs are important for sleep staging (**??**). Inspired by this, we design a Multi-level Feature Alignment method consisting of Epoch-level Feature Alignment and Sequence-level Feature Alignment. The alignment of sequence features $\mathbf{h}_j^{S_i} = \{h_{j,1}^{S_i}, h_{j,2}^{S_i}, ..., h_{j,L}^{S_i}\}$ could be divided to reducing marginal distribution on single epoch features and conditional distribution between epochs in a sleep sequence:

$$\begin{aligned}\mathrm{P}(\mathbf{h}_j^{S_i}) &= \mathrm{P}(h_{j,k}^{S_i})\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i}) \\ &= \frac{1}{L}\sum_{k=1}^{L} \mathrm{P}(h_{j,k}^{S_i})\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i}),\end{aligned} \tag{3}$$

where $\mathrm{P}(h_{j,k}^{S_i})$ is the marginal distribution of each epoch feature and $\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i})$ is the conditional distribution between each epoch feature and other epoch features in a sleep sequence. To summarize, **the feature alignment could be converted to the alignment of $\mathrm{P}(h_{j,k}^{S_i})$ and $\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i})$, which can be called Multi-level Feature Alignment.** We call the alignment of $\mathrm{P}(h_{j,k}^{S_i})$ as

Epoch-level Feature Alignment and the alignment of $\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i})$ as *Sequence-level Feature Alignment*.

Epoch-level Feature Alignment Some existing methods minimize the feature distribution divergence by minimizing the maximum mean discrepancy (MMD) (**???**) and other methods minimize the domain discrepancy by minimizing the second order correlation (**??**). Inspired by above, we utilize both the first-order statistics (expectation) and the second-order statistics (covariances) as the measure of distribution to minimize the domain discrepancy. We set F^i as the set of all features in the source domain S_i, so any $h_{j,k}^{S_i} \in F^i$. The first-order statistics discrepancy is computed through the following equation:

$$\mathcal{L}_{\text{first}} = \sum_{i \neq j} \|\mathrm{E}(F^i) - \mathrm{E}(F^j)\|_2, \tag{4}$$

where $\mathrm{E}(\cdot)$ is the expectation and $\|\cdot\|_2$ denotes the squared norm.

The second-order statistics discrepancy is computed through the following equation:

$$\mathcal{L}_{\text{second}} = \sum_{i \neq j} \|\mathrm{COV}(F^i) - \mathrm{COV}(F^j)\|_F^2, \tag{5}$$

where $\mathrm{COV}(\cdot)$ is the covariances matrix and $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm.

Combining the first-order statistics discrepancy and the second-order statistics discrepancy, we get the epoch-level discrepancy computed equation:

$$\mathcal{L}_{\text{epoch}} = \mathcal{L}_{\text{first}} + \mathcal{L}_{\text{second}}, \tag{6}$$

By minimizing $\mathcal{L}_{\text{epoch}}$, we can minimize the epoch-level domain discrepancy to align the marginal distribution $\mathrm{P}(h_{j,k}^{S_i})$.

Sequence-level Feature Alignment Our Sequence-level Feature Alignment is to align the conditional distribution $\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i})$. However, it is very difficult to align this distribution directly. In order to simplify this problem, we regard the conditional distribution $\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i})$ as the relationship between the feature of one epoch and the features of other epochs in the sleep sequence. Thus, we decide to use the Pearson correlation coefficients to represent the relationship between different epochs in a sequence and to align the expectation of the inter-epoch Pearson correlation coefficients in the sleep sequence from different source sleep datasets.

We set R_j^i as the correlation matrix of sleep sequence $\mathbf{h}_j^{S_i}$, and $\rho_{k,t}$ is the element of the k-th column and the t-th row in R_j^i. $\rho_{k,t}$ is computed through the following equation:

$$\rho_{k,t} = \frac{Cov(h_{j,k}^{S_i}, h_{j,t}^{S_i})}{\sqrt{Var(h_{j,k}^{S_i}) Var(h_{j,t}^{S_i})}}, \tag{7}$$

where $Cov(\cdot)$ is the covariances and $Var(\cdot)$ is the variance. After getting R_j^i, we can compute the expectation R^i of correlation matrix of source domain S_i through the following equation:

$$R^i = \frac{1}{N^{S_i}} \sum_{j=1}^{N^{S_i}} R_j^i, \tag{8}$$

where N^{S_i} denotes the number of sleep sequences in a source domain D^{S_i}.

Finally, we can compute the sequence-level discrepancy through the following equation:

$$\mathcal{L}_{\text{sequence}} = \sum_{i \neq j} \|R^i - R^j\|_F^2, \tag{9}$$

where $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm. By minimizing $\mathcal{L}_{\text{sequence}}$, we can indirectly align the conditional distribution $\mathrm{P}(\mathbf{h}_j^{S_i} \setminus \{h_{j,k}^{S_i}\}|h_{j,k}^{S_i})$ to minimize the sequence-level domain discrepancy.

Training

We design a classifier f to classify each epoch in a sequence into different sleep stages. The classifier f is a fully connected layer with a softmax function. We use the cross-entropy (CE) function as the loss function for the sleep staging task:

$$\mathcal{L}_{\text{classify}} = -\sum_{k=1}^{L} \sum_{l=1}^{N} y_{j,k,l}^{S_i} log(\hat{y}_{j,k,l}^{S_i}), \tag{10}$$

where $y_{j,k,l}^{S_i}$, the l-th element of $y_{j,k}^{S_i}$, denotes the probability that $x_{j,k}^{S_i}$ actually belongs to the l-th stage, and $\hat{y}_{j,k,l}^{S_i}$, the l-th element of $\hat{y}_{j,k}^{S_i}$, denotes the probability that $x_{j,k}^{S_i}$ is predicted to the l-th stage. Finally, we combine the loss functions for feature alignment and classifier:

$$\mathcal{L} = \mathcal{L}_{\text{classify}} + \lambda_1 \mathcal{L}_{\text{rec}} + \lambda_2 \mathcal{L}_{\text{epoch}} + \lambda_3 \mathcal{L}_{\text{sequence}}, \tag{11}$$

where $\mathcal{L}$ is the final loss function used in Eq. (1), λ_1,λ_2 and λ_3 is the coefficients.

Experiments

Datasets and Preprocessing

We evaluated the performance of SleepDG and other methods on **five different public sleep staging datasets: I. SleepEDFx** (**??**) consists of 197 PSG recordings, all of which were used for evaluation. **II. ISRUC** (**?**) consists of 126 whole-night PSG recordings, all of which were employed in this work. **III. SHHS** (**??**) consists of 5,791 PSG recordings. To keep the balance with other datasets, we used the first 150 recordings in our experiments. **IV. HMS** (**?**) includes 151 whole-night PSG recordings, and we used all the recordings in the experiments. **V. P2018** (**?**) consists of 944 whole-night PSG recordings. We employed the first 150 recordings in this work for the balance with other datasets. The summary of all the datasets is shown in Tab. 1. In order to ensure that SleepDG can adapt to datasets with different channels, we only selected single-channel EEG and EOG from each dataset.

SleepEDFx and SHHS were scored according to R&K standard (**?**), including W, N1, N2, N3, N4 and REM. We merged the N3 and N4 stages into a single stage N3 according to the latest AASM standard (**?**). Besides, for SleepEDFx, we only kept the PSGs starting from 30 minutes before to 30 minutes after the first and last non-wake epochs as recommended by **?**. All the sleep recordings used in our

Dataset	Sample Frequency (Hz)	Recordings (all)	Recordings (we choose)	EEG channel (we choose)	EOG channel (we choose)
I. SleepEDFx	100	197	197	Fpz-Cz	horizontal
II. ISRUC	200	126	126	F4-M1	E1-M2
III. SHHS	125	5,793	150	C4-M1	ROC-LOC
IV. HMC	256	151	151	F4-M1	E1-M2
V. P2018	200	995	150	C3-M2	E1-M2

Table 1: Summary of the datasets used in our experiments.

SD	II, III, IV, V		I, III, IV, V		I, II, IV, V		I, II, III, V		I, II, III, IV		Avg	
TD	I		II		III		IV		V			
Metrics	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1
DeepSleepNet	72.28	65.72	64.04	62.84	69.71	67.80	64.73	52.79	66.62	60.51	67.48	61.93
U-Time	72.51	65.84	65.13	63.71	70.58	68.10	64.53	51.68	67.35	61.07	67.82	62.28
AttnSleep	73.76	66.93	64.49	62.19	70.39	68.19	64.07	51.82	66.19	60.78	67.78	61.98
ResnetMHA	73.01	65.89	65.18	63.02	70.11	67.54	65.16	53.99	67.89	61.87	68.27	62.46
TinySleepNet	73.34	66.10	65.87	64.01	69.18	66.99	65.76	54.56	68.29	61.36	68.49	62.60
EnhancingCE	73.51	66.69	65.98	64.37	70.56	69.12	65.88	54.39	67.84	61.19	68.75	63.14
SalientSleepNet	73.92	67.59	65.44	63.22	69.93	68.16	66.36	55.49	68.89	63.13	68.91	63.52
SleepDG	**77.44**	**71.29**	**73.85**	**71.16**	**78.69**	**74.44**	**70.45**	**60.89**	**74.74**	**70.43**	**75.03**	**69.64**

Table 2: Performance comparison with existing non-DG methods for sleep staging. I is SleepEDFx, II is ISRUC, III is SHHS, IV is HMS, V is P2018.

experiments were band-pass filtered (0.3Hz–35Hz), resampled to 100Hz and normalized according to the Z-score standardization.

Settings

We take turns to select four ones from the five datasets as the source domains for training and set the left one as the unseen domain for testing, where the training data and testing data are from different datasets. We adopt *training-domain validation* (**?**) as the strategy of model selection, where each source domain is split into the training part and the validation part. Notably, **the unseen domain is inaccessible in the training procedure**.

We implemented SleepDG based on the PyTorch. The source code is publicly available[1]. The model is trained using the Adam optimizer with default settings, the learning rate is set to 1e-3 and the weight decay is set to 1e-4. The coefficients λ_1, λ_2 and λ_3 are all set to 0.5. The training epoch is 50, the mini-batch size is set to 32 and the dropout rate is 0.1. We set the length of sleep epoch sequence as $L = 20$ and the feature dimension as $d = 512$. Accuracy (ACC) and Macro-F1 score (MF1) are used as evaluation metrics. We trained the model on one machine with Intel Core i9 10900K CPU and eight NVIDIA RTX 3080 GPUs.

Results and Analysis

Compared with non-DG Methods for Sleep Staging Firstly, we compared with several non-DG methods for automatic sleep staging: **DeepSleepNet** (**?**) is a classical CNN-BiLSTM network for extracting local features and learning transition rules. **U-Time** (**?**) is a fully-CNN encoder-decoder architecture for time series segmentation applied to sleep staging. **AttnSleep** (**?**) is composed of a multi-resolution CNN and a multi-head self-attention with causal convolutions. **ResnetMHA** (**?**) uses a residual CNN to capture local features and a self-attention to model global temporal context. **TinySleepNet** (**?**) is a classical model based on CNN and RNN, with a smaller number of model parameters. **EnhancingCE** (**?**) captures both of local salient and global contextual features via two auxiliary tasks. **SalientSleepNet** (**?**) proposes a fully CNN based on the U^2-Net to detect multimodal salient waves.

We implemented these methods based on their public code and our DG settings. Tab. 2 shows the performance comparison with the methods. **SleepDG achieves the state-of-the-art performance in all the datasets** (75.03% in average ACC and 69.64% in average MF1), proving that SleepDG can better solve the generalization problem of unseen domain compared with traditional sleep staging methods even if subject population and signal channels of source domains are different from those of the unseen domain. The DeepSleepNet performs the worst, about 7.6% lower in average ACC and 7.7% lower in average MF1 than SleepDG, indicating that classical model is difficult to deal with DG scenarios. TinySleepNet performs a little better than DeepSleepNet (68.49% v.s. 67.48% in average ACC and 62.60% v.s. 61.93% in average MF1). It indicates that a smaller number of model parameters may alleviate overfitting issue in DG scenarios. U-Time, AttnSleep, ResnetMHA and EnhancingCE cannot obtain a significant improvement in performance compared with the classical method DeepSleep-

[1]https://github.com/wjq-learning/SleepDG

SD	II, III, IV, V		I, III, IV, V		I, II, IV, V		I, II, III, V		I, II, III, IV		Avg	
TD	I		II		III		IV		V			
Metrics	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1
BASE	73.89	67.02	65.63	63.85	70.34	68.49	65.10	54.47	68.83	62.85	68.76	63.34
MMD	75.41	70.06	69.39	66.84	75.09	70.60	68.09	57.68	70.71	65.00	71.74	66.04
CORAL	75.38	70.70	70.06	68.57	74.83	72.18	69.45	58.33	71.62	67.81	72.27	67.52
IRM	75.27	70.21	70.24	68.42	74.84	71.81	68.48	56.43	71.60	67.88	72.09	66.95
DANN	74.71	69.56	69.30	67.69	72.69	70.35	67.49	56.17	71.01	67.40	71.12	66.23
SleepDG	**77.44**	**71.29**	**73.85**	**71.16**	**78.69**	**74.44**	**70.45**	**60.89**	**74.74**	**70.43**	**75.03**	**69.64**

Table 3: Performance comparison with existing DG methods for sleep staging. I is SleepEDFx, II is ISRUC, III is SHHS, IV is HMS, V is P2018.

SD	II, III, IV, V		I, III, IV, V		I, II, IV, V		I, II, III, V		I, II, III, IV		Avg	
TD	I		II		III		IV		V			
Metrics	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1	ACC	MF1
BASE	73.89	67.02	65.63	63.85	70.34	68.49	65.10	54.47	68.83	62.85	68.76	63.34
AE	74.34	68.08	66.84	64.82	72.21	70.67	65.76	55.38	69.48	63.22	69.73(**+0.97**)	64.43(**+1.09**)
EA	75.88	71.00	71.56	69.69	75.63	72.10	66.98	56.67	71.62	67.61	72.33(**+3.57**)	67.41(**+4.07**)
SA	74.41	68.56	69.47	68.84	72.09	69.60	68.98	58.51	70.71	67.00	71.13(**+2.37**)	66.50(**+3.16**)
AE+EA	75.89	70.90	72.34	70.18	76.58	73.23	68.28	57.60	72.81	68.34	73.18(**+4.42**)	68.05(**+4.71**)
AE+SA	74.38	70.16	70.30	69.13	75.65	72.99	67.23	55.25	71.74	68.44	71.86(**+3.10**)	67.19(**+3.85**)
SleepDG	**77.44**	**71.29**	**73.85**	**71.16**	**78.69**	**74.44**	**70.45**	**60.89**	**74.74**	**70.43**	**75.03(+6.27)**	**69.64(+6.30)**

Table 4: Performance comparison with ablated methods. I is SleepEDFx, II is ISRUC, III is SHHS, IV is HMS, V is P2018.

Net, indicating that it is difficult to achieve a significant performance improvement in DG scenarios by only modifying the network structures. SalientSleepNet achieves a significant performance improvement on specific dataset (**?**), but it only performs about 1.5% higher in average ACC and 1.6% higher in average MF1 compared with DeepSleepNet, indicating that a complex neural network design cannot solve the problem of domain shift well. Compared with these traditional non-DG methods, SleepDG can learn domain-invariant features through domain alignment, and further improve generalization ability to unseen domains.

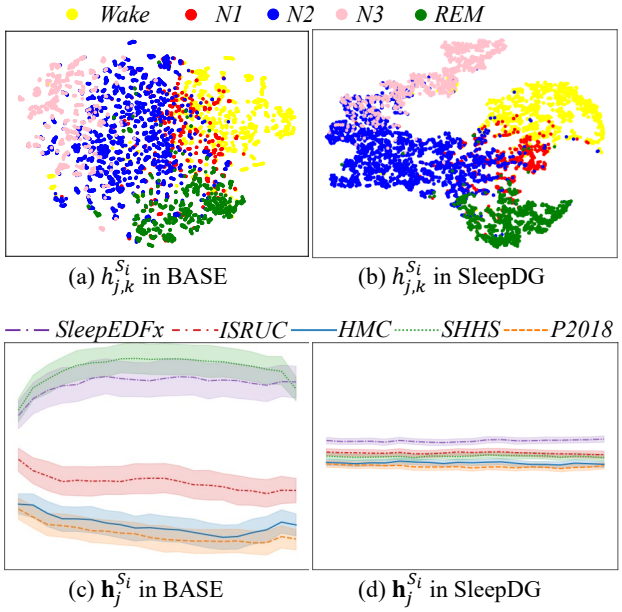
Compared with DG Methods for Sleep Staging We compared SleepDG with BASE and other classical DG methods in sleep staging: **BASE** can be regarded as SleepDG which minimizes $\mathcal{L}_{\text{classify}}$ but without any reconstruction and DG loss. **MMD** (Maximum Mean Discrepancy) (**??**) is a DG method matching the MMD of feature distributions. **CORAL** (**?**) matches the covariance of feature distributions. **IRM** (Invariant Risk Minimization) (**?**) enforces the optimal classifier to be the same across all domains. **DANN** (Domain-Adversarial Neural Networks) (**?**) employs an adversarial network to match feature distributions.

We implemented these DG methods based on BASE and DeepDG toolkit (**?**), and the performance comparison is shown in Tab. 3. SleepDG achieves the best performance compared with other DG methods. BASE performs the worst (68.76% in average ACC and 63.34% in average MF1), indicating that a model without DG loss cannot minimize the domain shift among different domains. MMD performs better compared with BASE, about 3.0% higher in average ACC and 2.7% higher in average MF1, suggesting minimizing maximum mean discrepancy can extract domain-invariant features to a degree. CORAL performs the best in existing DG methods (72.27% in average ACC and 67.52% in average MF1), which indicates that feature alignment based on second-order statistics (covariance) can achieve good cross-domain generalization. IRM performs better than MMD but worse than CORAL (72.09% in average ACC and 66.95% in average), indicating that invariant risk minimization has no obvious advantage over feature alignment methods. The domain-adversarial model of DANN achieves the least improvement compared with BASE (71.12% v.s. 68.76% in average ACC). It matches the finding in **?**: although domain adversarial training often achieves better performance in DA, it is difficult to make a significant improvement in DG. Compared with these classical DG methods, SleepDG can learn domain-invariant features better by combining Epoch-level Feature Alignment and Sequence-level Feature Alignment.

Ablation Study To investigate the effectiveness of signal reconstruction and feature alignment in SleepDG, we conducted an ablation study. Here, we also take **BASE** as our baseline model. The ablated methods are **AE** (Base + Autoencoder), **EA** (Base + Epoch-level Feature Alignment), **SA** (Base + Sequence-level Feature Alignment), **AE+EA** (Base + Autoencoder + Epoch-level Feature Alignment) and **AE+SA** (Base + Autoencoder + Sequence-level Feature

Alignment). The results are shown in Tab. 4. AE performs a little better than BASE about 0.97% higher in average ACC and 1.09% higher in average MF1, indicating that AE-based representation learning can learn generalizable features to a certain extent. EA contributes greatly to the sleep staging, obtaining a great performance improvement compared with BASE, 3.57% and 4.07% higher in average ACC and average MF1, respectively. AE+EA also performs well, obtaining 4.42% and 4.71% higher in average ACC and average MF1 than Base. It indicates the Epoch-level Feature Alignment can extract domain-invariant epoch features well. Meanwhile, compared with Base, SA performs about 2.37% higher in average ACC and 3.16% higher in average MF1, and AE+SA performs about 3.10% and 3.85% higher, indicating that Sequence-level Feature Alignment can align context features. SleepDG can further improve the ability to learning domain-invariant features in both epoch and sequence level, improving the performance of 6.27% in average ACC and 6.30% in average MF1 compared with BASE. Notably, SleepDG achieves the best performance across all the unseen domain settings, indicating that SleepDG has a relatively stable generalization performance.

Feature Visualization We conducted a feature visualization to further show the effectiveness of SleepDG as Fig. 2 shows. Fig. 2(a)(b) show epoch features visualization of different domains from BASE and SleepDG. The different colors represent different sleep stages. Here, we visualize the features by t-SNE (**?**). Compared with the feature representations in Fig. 2(a), in Fig. 2(b) the feature representations in the same sleep stage obviously form a cluster and the ones in different sleep stages nicely separate with each other after we adopt our alignment methods, even though these features are totally from different domains. It intuitively indicates that our SleepDG can learn domain-invariant feature representations which are independent on the specific domains. Meanwhile, it further proves that the feature representations obtained by SleepDG are more distinguishable for automatic sleep staging. Fig. 2(c)(d) show sequence feature visualization of BASE and SleepDG. We use PCA to reduce the dimension of every $h_{j,k}^{S_i}$ in sequence features $\mathbf{h}_j^{S_i}$ to one dimension and draw a linear regression line plot for each domain. Obviously, the sequence features from SleepDG form a closer cluster compared with those from BASE, which intuitively indicates that our Sequence-level Feature Alignment is effective to align the inter-epoch relationship.



(a) $h_{j,k}^{S_i}$ in BASE (b) $h_{j,k}^{S_i}$ in SleepDG

(c) $\mathbf{h}_j^{S_i}$ in BASE (d) $\mathbf{h}_j^{S_i}$ in SleepDG

Figure 2: Feature visualization of BASE and SleepDG.

Conclusion

We propose a novel task of generalizable sleep staging to solve the severe generalization problem of automatic sleep staging in clinic. In order to improve the generalization ability of sleep staging methods, we propose a novel DG-based framework. Considering both of the local salient features within each sleep epoch and sequential features among different epochs, we develop a Multi-level Feature Alignment method which focuses on both epoch-level and sequence-level feature alignment. Specifically, we design an Epoch-level Feature Alignment method aligning both mean and covariance of single-epoch feature distribution among different domains to learn epoch-level domain-invariant features. We also design a Sequence-level Feature Alignment method to extract sequence-level domain-invariant features, by minimizing the discrepancy of Pearson correlation coefficients of sleep sequence among different domains. Our method is validated on five public sleep datasets with DG settings, achieving the state-of-the-art performance.

Acknowledgments

This work was supported by the Key Program of the Natural Science Foundattion of Zhejiang Province, China (No. Z24F020009), STI 2030 Major Projects (No. 2021ZD0200400) and Natural Science Foundation of China (No. 61925603). The corresponding authors are Dr. Sha Zhao and Dr. Gang Pan. distinctio inventore nesciunt, culpa natus quasi odit tempora autem perspiciatis porro accusamus, rerum fuga nihil earum odio.Expedita hic amet, id adipisci doloribus velit optio exercitationem.Totam accusamus beatae nisi eaque maxime, vero maxime necessitatibus optio expedita blanditiis amet inventore fugiat vel dolor, odio ullam natus, earum numquam nulla inventore?Ipsa exercitationem dolor provident cumque aperiam repellat ratione porro sequi saepe accusantium, vitae nam veniam unde distinctio et incidunt sapiente, eveniet odio velit nulla explicabo nam quaerat eius perspiciatis eum reiciendis, ducimus optio neque natus eaque pariatur quas consequatur.Vero doloremque cupiditate inventore tempora architecto consectetur facere consequuntur voluptas numquam, reprehenderit molestias vel aliquam porro veritatis.Nihil eos sed vero rerum, odit perspiciatis quidem, rerum iure eum adipisci dolores omnis vel quidem accusamus consequatur, alias quas ut omnis nobis ducimus.Quis autem nostrum aperiam vel aspernatur, temporibus consectetur dolorem iste incidunt eaque laudantium sed, ratione sequi provident hic nesciunt

accusantium eum ullam qui modi esse alias, iusto eius officiis accusamus provident nobis itaque eum animi vel?