

PAPER

Data augmentation for enhancing EEG-based emotion recognition with deep generative models

To cite this article: Yun Luo *et al* 2020 *J. Neural Eng.* **17** 056021

View the [article online](#) for updates and enhancements.

You may also like

- [Learning-based local-to-global landmark annotation for automatic 3D cephalometry](#)
Hye Sun Yun, Tae Jun Jang, Sung Min Lee *et al.*
- [Investigating EEG-based functional connectivity patterns for multimodal emotion recognition](#)
Xun Wu, Wei-Long Zheng, Ziyi Li *et al.*
- [Data-driven molecular design for discovery and synthesis of novel ligands: a case study on SARS-CoV-2](#)
Jannis Born, Matteo Manica, Joris Cadow *et al.*



PAPER

Data augmentation for enhancing EEG-based emotion recognition with deep generative models

RECEIVED
4 June 2020REVISED
24 July 2020ACCEPTED FOR PUBLICATION
4 September 2020PUBLISHED
12 October 2020Yun Luo¹ , Li-Zhen Zhu¹, Zi-Yu Wan¹ and Bao-Liang Lu^{1,2,3,4,5} ¹ Center for Brain-like Computing and Machine Intelligence, Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, People's Republic of China² Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, People's Republic of China³ Brain Science and Technology Research Center, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, People's Republic of China⁴ Qing Yuan Research Institute, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, People's Republic of ChinaE-mail: blu@sjtu.edu.cn**Keywords:** emotion recognition, EEG, GAN, VAE, deep generative model, data augmentation, deep neural network**Abstract**

Objective. The data scarcity problem in emotion recognition from electroencephalography (EEG) leads to difficulty in building an affective model with high accuracy using machine learning algorithms or deep neural networks. Inspired by emerging deep generative models, we propose three methods for augmenting EEG training data to enhance the performance of emotion recognition models. *Approach.* Our proposed methods are based on two deep generative models, variational autoencoder (VAE) and generative adversarial network (GAN), and two data augmentation ways, full and partial usage strategies. For the full usage strategy, all of the generated data are augmented to the training dataset without judging the quality of the generated data, while for the partial usage, only high-quality data are selected and appended to the training dataset. These three methods are called conditional Wasserstein GAN (cWGAN), selective VAE (sVAE), and selective WGAN (sWGAN). *Main results.* To evaluate the effectiveness of these proposed methods, we perform a systematic experimental study on two public EEG datasets for emotion recognition, namely, SEED and DEAP. We first generate realistic-like EEG training data in two forms: power spectral density and differential entropy. Then, we augment the original training datasets with a different number of generated realistic-like EEG data. Finally, we train support vector machines and deep neural networks with shortcut layers to build affective models using the original and augmented training datasets. The experimental results demonstrate that our proposed data augmentation methods based on generative models outperform the existing data augmentation approaches such as conditional VAE, Gaussian noise, and rotational data augmentation. We also observe that the number of generated data should be less than 10 times of the original training dataset to achieve the best performance. *Significance.* The augmented training datasets produced by our proposed sWGAN method significantly enhance the performance of EEG-based emotion recognition models.

1. Introduction

Emotion plays a significant role in how people think, behave, and communicate. Artificial emotional intelligence, which is also known as emotion AI or affective computing, focuses on developing devices and systems that can automatically recognize human emotion and has attracted considerable attention very

recently (Somers 2019, Smith and Burke 2019). For example, integrating emotion assessment in human-computer interaction systems with emotion recognition can make machines more intelligent and provide more humanized interactions. Moreover, studies have shown that some mental diseases, such as depression and autism, are relevant to emotions (Bocharov *et al* 2017). The introduction of emotion AI to these studies can create a high potential for treating psychiatric diseases. Because emotion AI has many

⁵ Author to whom any correspondence should be addressed.

potential applications, attention is being focused on the possibility of recognizing emotions from different behavioral cues, such as facial expression (Liu *et al* 2017), posture (Garber-Barron and Si 2012), voice (Tanja *et al* 2009), and neurophysiological signals (Samara *et al* 2017). Among these signals, electroencephalography (EEG) has been demonstrated as one of the most reliable signals due to its high accuracy and objective benefits. In recent years, EEG-based emotion recognition has attracted widespread attention from academics and industries (Wang *et al* 2014, McFarland *et al* 2016, Meneses Alarcro and Fonseca 2017, Craik *et al* 2019, Fourati *et al* 2020). Researchers have made considerable progress in feature extraction and model construction. However, these studies are faced with a problem: the lack of training data.

Compared with visual and audio signals, which can be easily accessed from standard datasets, data acquisition is still one of the bottlenecks in EEG-based emotion recognition tasks. There are mainly five reasons: a) The price of EEG acquisition devices for research is quite high. b) These experiments cannot last for a long time because the subjects may feel uncomfortable wearing EEG acquisition devices. c) The raw EEG data are usually mixed with noise and various artifacts, and researchers have to discard some bad channels and data, which aggravates the data scarcity problem. d) It is difficult to collect precisely labeled data since the subjects may not evoke emotion well in emotion recognition experiments. e) There are only a few public EEG-based emotion recognition datasets, such as SEED¹ (Zheng and Lu 2015, Zheng, Liu, Lu, Lu and Cichocki 2019), DEAP² (Koelstra *et al* 2012), DREAMER (Katsigianis and Ramzan 2018), MAHNOB-HCI³ (Soleymani *et al* 2012), and MPED (Song *et al* 2019). Moreover, the scales of these datasets are much smaller than those of public image datasets (e.g. ImageNet). These factors limit the quantity of labeled training data for EEG-based emotion recognition and hinder the performance of emotion recognition models trained by machine learning algorithms and deep neural networks.

It is common sense that a machine learning model will be more accurate when it can access more training data. For example, the release of the trillion-word corpus by Google improves text-based models (Halevy *et al* 2009). Machine learning models can be more robust and reliable when learning more effective features from sufficient training data, especially for deep learning models that need a vast quantity of training data. Deep learning models have recently achieved remarkable results in the fields of computer vision, speech recognition, and natural language processing

due to the accessibility of large datasets (Zhang *et al* 2018).

In the field of EEG-based emotion recognition, Zheng and Lu used deep neural networks to recognize three emotions and reached a compelling accuracy (Zheng and Lu 2015). In their work, they only applied a two-layer deep belief network. The achievements in the image, speech, and natural language processing fields indicate that there is considerable room for further studying the problem of EEG-based emotion recognition by leveraging the ability of deeper neural networks. However, compared with shallow layer models, deep-layer models use more parameters and require a large number of labeled training data to explore the potentials of deep neural networks. Consequently, the primary issue that should be addressed in EEG-based emotion recognition is to acquire sufficient and high-quality training data.

Generating artificial data by applying a transformation from the original data is one of the conventional solutions to solving the data scarcity problem. This approach is called data augmentation. Recently, various data augmentation methods have been applied to generate EEG data (Krell and Kim 2017, Lotte 2015, Wang *et al* 2018). Some researchers have generated EEG data by applying a geometric transformation to the original data and reported the performance of classifiers improved by adding the generated data. Other researchers have focused on using deep generative models to generate artificial EEG data (Hartmann *et al* 2018). Compared with signal-level transformation through geometric transformation, the deep generative model could learn the representation of the real distribution at a deeper level. However, the performance of the classifier after data augmentation were not demonstrated (Hartmann *et al* 2018). In our previous study, we generated realistic-like EEG features by taking advantage of GANs (Luo and Lu 2018). Then, we compared the performance of affective models without and with appending the generated data to the training dataset. The experimental results demonstrated that the GAN-based data augmentation method could improve the performance of affective models. In this paper, we further explore the generative methods based on the above achievement.

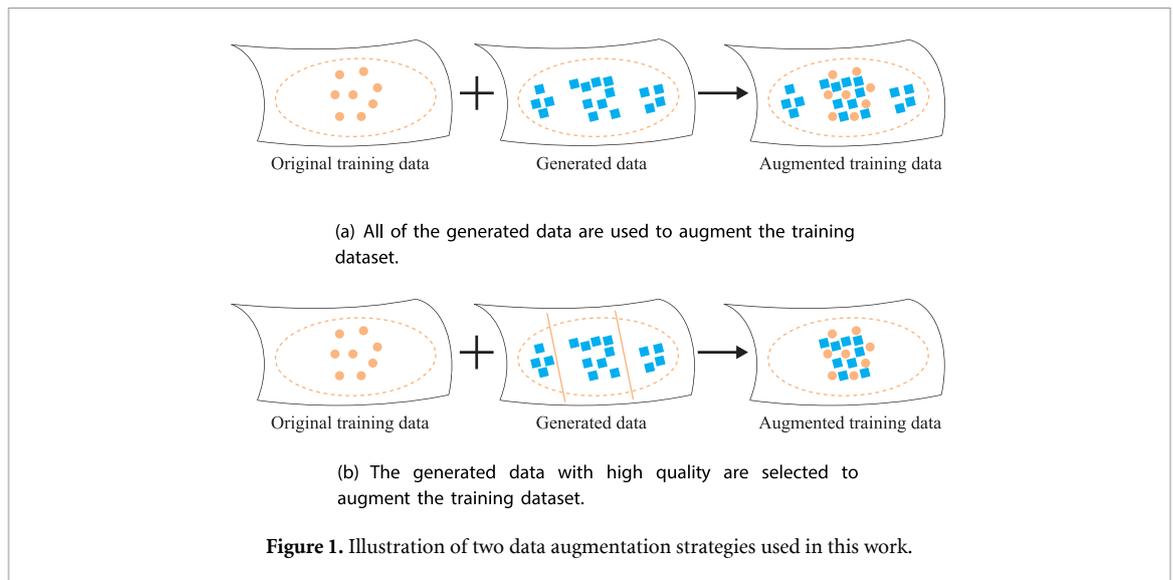
It is common for classifiers to handle the high-level features of EEG data in EEG-based emotion recognition tasks. Therefore, this work focuses on generating power spectral density (PSD) and differential entropy (DE) features, which are two commonly used features in emotion recognition tasks (Zheng and Lu 2015, Duan *et al* 2013, Yang *et al* 2018).

The work includes two emerging deep generative models: variational autoencoder (VAE) (Kingma and Welling 2014) and Wasserstein generative adversarial network (WGAN) with gradient penalty (Arjovsky *et al* 2017, Gulrajani *et al* 2017). We propose two

¹<http://bcmi.sjtu.edu.cn/~seed/index.html>.

²<http://www.eecs.qmul.ac.uk/mmv/datasets/deap/>.

³<https://mahnob-db.eu/hci-tagging/>.



data augmentation strategies: full usage of generated data and partial usage of generated data. As illustrated in figure 1, the basic ideas behind the full usage strategy and partial usage strategy are to use all of the generated data and select part of the generated data. Since we cannot guarantee that all of the generated data have high qualities, it is important for us to decide how to use the generated data. For the full usage strategy, we propose conditional Wasserstein GAN (cWGAN) to control the category of the generated data. Then, we append all of the generated data to the original training dataset without considering their quality.

For the partial usage strategy, we propose two methods called selective VAE (sVAE) and selective WGAN (sWGAN) to generate data. In these two methods, the generated data are unlabeled. And we choose the generated data with high classification confidence and append the selected data to the original training dataset. Unlike images, the generated EEG features are high-dimensional data and are intractable for humans to judge the quality of the generated data. Therefore, these two methods are based on a simple idea, and the generated data are regarded as high quality when they are classified with high classification confidence by a classifier trained by the original dataset. We use two conventional pattern classifiers, SVMs and deep neural networks with shortcut layers, to train affective models on two public EEG datasets widely used for emotion recognition: SEED and DEAP.

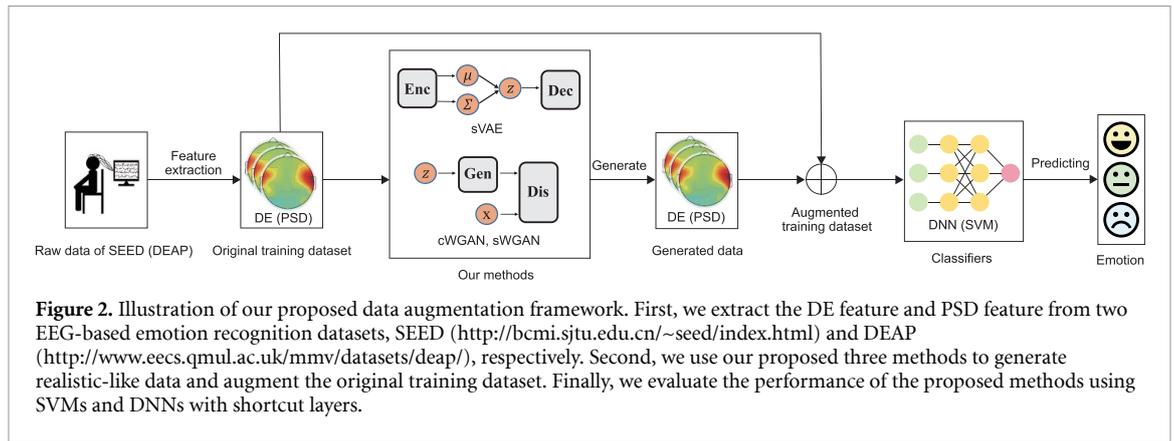
For a comparison study, we introduce three conventional data augmentation methods for EEG-based emotion recognition: conditional VAE (cVAE) (Kingma and Welling 2014), which adopts a similar generated strategy as cWGAN, Gaussian noise method (Gau), which augments the datasets by adding Gaussian noise to the original data (Wang et al 2018), and rotational data augmentation method

(RDA), which generates new data by applying a geometric rotation to the original data (Krell and Kim 2017). We perform a systematic experimental study to compare the proposed methods with these conventional methods. We use 5-fold cross-validation to measure the classification performance of different augmented methods. The proposed framework is illustrated in figure 2.

The main contributions of this paper lie in the following aspects:

1. To the best of our knowledge, we adopt deep generative methods to augment EEG training data for emotion recognition for the first time.
2. We propose three methods for generating EEG data based on different generative methods and two different strategies for using the generated data.
3. We carry out a systematic comparison between different features, different generative methods and different classifiers on two EEG datasets. The experimental results demonstrate that our proposed methods could make the affective models have better performance on EEG-based emotion recognition.

The rest of the paper is organized as follows. Section 2 provides an overview of related work on generative methods, data augmentation methods for EEG-based emotion recognition, and a brief introduction to deep neural networks. In section 3, we introduce different methods in detail. Section 4 describes the two datasets, SEED and DEAP, and presents the details of our experimental settings. A systematic comparison between different methods and the efficiency of our proposed methods by conducting a series of data augmentation experiments is presented in section 5. Finally, in section 6, we present conclusions about our work.



2. Related work

In this section, we briefly introduce relevant work on EEG-based emotion recognition, deep generative methods, data augmentation methods, and deep neural networks.

2.1. EEG-based emotion recognition

EEG-based emotion recognition has received considerable attention. Mühl *et al* introduced affective factors into traditional brain-computer interfaces (BCIs) and presented the definition of affective brain-computer interfaces (aBCIs) (Mühl *et al* 2014). Alarcao and Fonseca surveyed different EEG-based emotion recognition methods and compared the main aspects involved in the recognition process, including stimuli, feature extraction methods, and classifiers (Alarcao and Fonseca 2017). Jenke *et al* reviewed feature extraction and selection methods for 33 EEG-based emotion recognition studies (Jenke *et al* 2014). Petrantonakis and Hadjileontiadis presented a novel EEG-based feature extraction technique by employing higher-order crossings analysis (Petrantonakis and Hadjileontiadis 2010).

Koelstra *et al* developed a publicly available EEG-based emotion dataset called DEAP by recruiting 32 subjects to watch 40 music videos. Zheng and Lu required 15 subjects to watch 15 selected Chinese movie clips to elicit three emotions: happy, sad, and neutral (SEED dataset) (Zheng and Lu 2015). Then, they performed a systematic comparison between various feature extraction, feature selection, feature smoothing and classification methods in a three-category EEG-based emotion recognition task and showed the stable patterns of EEG in this task (Zheng *et al* 2019).

2.2. Deep generative methods

Generative models aim to learn the data distribution of a given dataset using unsupervised learning to generate new data with some variations and have been widely studied in the field of machine learning. Recent advances in parameterizing these models using deep neural networks have allowed them

to scale to diverse data, including images, text, and speech. Two of the most promising and efficient deep generative models are the variational autoencoder (VAE) (Kingma and Welling 2014) and generative adversarial network (GAN) (Goodfellow *et al* 2014).

As a variation in the autoencoder, VAE aims at generating new data only based on the given data (Kingma and Welling 2014). It solves the variational inference problem that maximizes the marginalized data likelihood by using a generative network (encoder) and a recognition network (decoder). At the end of the training, the encoder can generate realistic-like data. VAEs have shown great potential in generating different data (Salimans *et al* 2015, Kulkarni *et al* 2015, Gregor *et al* 2015).

As an emerging topic, GAN has attracted growing interest. The idea of GAN is to sample noise from distributions such as Gaussian and transform them into real data distributions. GANs are based on a min-max game theory that aims to find the Nash equilibrium between the two components, generator and discriminator. After the adversarial process, the generator can produce high-quality faked data. GANs have shown promise in generating realistic-like data in specific fields. Ledig *et al* proposed SRGAN for image superresolution (Ledig *et al* 2017). Wu *et al* proposed 3D-GAN to generate 3D objects from a probabilistic space using volumetric convolutional networks and generative adversarial networks (Wu *et al* 2016). In addition, GANs have also been applied to the generation of dialogue (Li *et al* 2017), electronic health records (EHRs) (Choi *et al* 2017), and polyphonic music (Mogren 2016). Considering that the original GAN has no control over modes of the generated data, Mirza and Osindero added the label as an additional parameter to the generator and the discriminator to control the category of the generated data (Mirza and Osindero 2014).

Although GANs have demonstrated great generation abilities, they have some problems, such as non-convergence, mode collapse, and diminished gradient. Chief among them is training stability (non-convergence), which is mainly caused by the adversarial game. Some pioneering works focus on fixing this

problem. Radford *et al* reported some network architecture recommendations about GANs and designed a sophisticated network called DCGAN (Radford *et al* 2016). Their work made a great contribution to solving the instability problem of GANs' training process. However, DCGAN is designed for image generation and requires specific design techniques, which limits it to scale to other fields. For this reason, other researchers have focused on altering the structure (Berthelot *et al* 2017) or the loss function (Qi 2017) of the original GAN to ensure training stability. Wasserstein GAN (WGAN) is one of the most dramatic attempts to handle this problem (Arjovsky *et al* 2017). Arjovsky *et al* regarded the mini-max game as minimizing the Wasserstein distance between the two distributions and replaced the original loss function of GAN by the Wasserstein distance. Their work significantly improved the stability of GAN training while maintaining the generation ability of GANs. In addition, WGAN requires no extra sophisticated network designation and can be easily applied to the generation of different signals, such as EEG. Based on WGAN, Gulrajani *et al* proposed using a gradient penalty in the training, which improved the performance of WGAN (Gulrajani *et al* 2017).

2.3. Data augmentation

Data augmentation aims at generating new data of the given dataset by applying transformations to the original data while preserving the label (van Dyk and Meng 2001). This method is commonly applied to reduce overfitting and improve classification performance (Krizhevsky *et al* 2012) since the generated data have a similar data distribution to the original data and can be used to increase the quantity of training data. In the field of image classification with small data size, this technique has been successfully adopted (Perez and Wang 2017). It is common to generate additional images by applying different distortions, scaling, or moving window/pixel shifts to the real images (Simard *et al* 2003). A similar technique has also been adopted to generate EEG signals. Krell and Su proposed rotational distortions that were similar to affine/rotational distortions of images to generate artificial EEG signals (Krell and Kim 2017). Lotte generated artificial EEG trials by the relevant combinations and distortions of the original trials (Lotte 2015). Wang *et al* generated EEG features by directly adding different Gaussian noises to the original feature and applied deep neural networks to verify the effect (Wang *et al* 2018). All of the abovementioned methods reported that the performance of the classifiers was improved by data augmentation.

Some pioneering works have focused on augmenting data by GANs, which demonstrated great generative ability. Zheng *et al* adopted DCGAN to generate images and used artificial images for person

reidentification tasks (Zheng *et al* 2017). Their results presented the feasibility of GAN-based data augmentation. They also reported that the classifier was less prone to overfitting by adding generated training samples. By applying a CycleGAN to augment the training dataset, Zhu *et al* improved the classification accuracy of the emotion recognition task based on images (Zhu *et al* 2018). For EEG signal generation, Hartmann *et al* proposed EEG-GAN to generate raw EEG signals (Hartmann *et al* 2018). In their work, they presented a series of evaluation metrics to demonstrate the potential for GANs to generate EEG data. However, they did not report the performance of the classifier when adding the generated EEG data to the training dataset. In our previous work, we extended the GAN-based augmentation method to EEG-based emotion recognition (Luo and Lu 2018). The experimental results demonstrated the efficiency of our data augmentation method for EEG-based emotion recognition.

2.4. Deep neural networks

Deep neural networks have been widely applied to the fields of computer vision (Hinton and Salakhutdinov 2006, Simonyan and Zisserman 2014), natural language processing (Mikolov *et al* 2013) and speech recognition (Dahl *et al* 2011). Although deep neural networks obtain exciting results in these fields, they still suffer from problems such as the curse of depth. It is difficult to train a neural network effectively with too many layers. To solve this problem, He *et al* proposed a residual learning framework called Resnet (He *et al* 2016), which had shortcuts between layers to transform the information. Inspired by this, we apply the deep neural network (DNN) with shortcut layers as one of our classifiers.

3. Method

In this section, we first give a brief introduction to VAE and WGAN. Then, we present our three deep generative models, cWGAN, sVAE, and sWGAN. Next, we describe three conventional data augmentation methods, cVAE, Gau, and RDA. Finally, we briefly describe DNN with shortcut layers.

3.1. VAE

The VAE is a latent variable generative model that consists of an encoder and a decoder. This model combines variational inference with the conventional autoencoder framework. The encoder encodes x into a latent representation space z , where x represents a real datapoint and has weights and biases λ . We denote the encoder $q_{\lambda}(z|x)$. The decoder outputs the probability distribution of real data given the latent representation z . It has weights and biases ϕ , which is denoted by $p_{\phi}(x|z)$.

The generative model aims to maximize the probability of each x in the training set according to

$$p(x) = \int p(x|z)p(z)dx. \quad (1)$$

However, this integral requires exponential time to compute. In practice, $p(x|z)$ will be nearly zero for most z , which contributes almost nothing to estimate $p(x)$. The VAE attempts to sample z , which is likely to produce x , by approximating the posterior $p(z|x)$ with $q_\lambda(z|x)$. It uses the Kullback-Leibler (KL) divergence, which measures the distance between two distributions:

$$\begin{aligned} KL(q_\lambda(z|x)||p(z|x)) &= \mathbb{E}_{z \sim q}[\log(q_\lambda(z|x)) - \log(p(z|x))] \\ &= \mathbb{E}_{z \sim q}[\log(q_\lambda(z|x)) - \log(p(x|z)) \\ &\quad - \log(p(z))] + \log(p(x)). \end{aligned} \quad (2)$$

The goal of KL divergence is to find the parameter λ to minimize this divergence. However, it is still impossible to compute the KL divergence directly since $p(x)$ appears in the formula, which is intractable as mentioned above. We can define the following function:

$$\begin{aligned} ELBO &= -\mathbb{E}_{z \sim q}[\log(q_\lambda(z|x)) - \log(p(x|z)) \\ &\quad - \log(p(z))], \end{aligned} \quad (3)$$

where $ELBO$ represents the evidence lower bound. Combining equations (3) with the KL divergence and rewrite, $p(x)$ can be rewritten as

$$\log(p(x)) = ELBO + KL(q_\lambda(z|x)||p(z|x)). \quad (4)$$

Note that the KL divergence is always greater than or equal to zero according to Jensen's inequality. Therefore, minimizing the KL divergence is equivalent to maximizing $ELBO$.

Now, we can decompose the $ELBO$ into a sum where each term depends on a single datapoint since no datapoint shares its latent z with another datapoint in VAE. We can write the $ELBO_i$ for a single datapoint i (the i^{th} datapoint) as

$$\begin{aligned} ELBO_i &= -\mathbb{E}_{z \sim q}[\log(q_\lambda(z|x_i)) - \log(p(x_i)|z)) \\ &\quad - \log(p(z))] + \log(p(x_i)) \\ &= \mathbb{E}_{z \sim q}[\log(p_\lambda(x_i|z))] - KL(q_\lambda(z|x_i)||p(z)), \end{aligned} \quad (5)$$

where the first term is the expected log-likelihood and the second term is the negative KL divergence between the encoder distributions $q_\lambda(z|x_i)$ and $p(z)$. The first term forces the decoder to learn to reconstruct the data from latent representation, and poor reconstruction results in a large cost in this loss function. The second term can also be viewed as

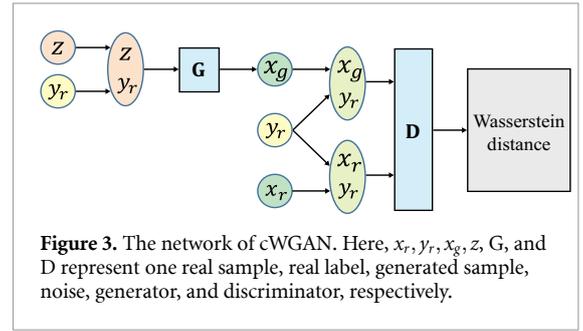


Figure 3. The network of cWGAN. Here, x_r, y_r, x_g, z, G , and D represent one real sample, real label, generated sample, noise, generator, and discriminator, respectively.

a regularizer, which measures how much information is lost when using $q_\lambda(z|x_i)$ to represent $p(z)$. The encoder receives a penalty if it outputs latent representations z that are different from those from $p(z)$. This term maintains the diversity of the latent representation.

In VAE, the choice of $p(x|z)$ is often a Gaussian distribution. Then, the first term of $ELBO_i$ can also be viewed as the reconstruction loss. The VAE assumes $p(z) = N(0, I)$ and $q_\lambda(z|x_i) = N(\mu(x_i), \Sigma(x_i))$, where N represents a Gaussian distribution. Therefore, the second term of $ELBO_i$ can be formalized as:

$$\begin{aligned} KL(q_\lambda(z|x_i)||p(z)) &= KL(N(\mu(x_i), \Sigma(x_i))||N(0, I)) \\ &= \frac{1}{2}(\text{tr}(\Sigma(x_i)) + \mu(x_i)^T \mu(x_i) \\ &\quad - k - \log(\det(\Sigma(x_i))))), \end{aligned} \quad (6)$$

where k is the dimension of the Gaussian distribution and $\text{tr}(x_i)$ is the trace function. We define $\Sigma(x_i)$ as a diagonal matrix, so the formula can be rewritten as

$$\begin{aligned} KL(q_\lambda(z|x_i)||p(z)) &= \frac{1}{2} \sum_k [\Sigma(x_i) + \mu^2(x_i) - 1 \\ &\quad - \log \Sigma(x_i)]. \end{aligned} \quad (7)$$

In practice, we use $\log \Sigma(x_i)$ instead of $\Sigma(x_i)$ since it is more numerically stable to take the exponent. Hence, the final goal of VAE is

$$\begin{aligned} \max_{\lambda, \phi} ELBO &= \sum_i ELBO_i \\ &= \sum_i \sum_k [(x_i - \hat{x}_i)^2 + \frac{1}{2}(\Sigma(x_i) + \mu^2(x_i) \\ &\quad - 1 - \log \Sigma(x_i))], \end{aligned} \quad (8)$$

where \hat{x}_i is the reconstructed data, and $\mu(x_i)$ and $\log \Sigma(x_i)$ are both calculated by the neural network.

3.2. WGAN

A typical GAN consists of two competing parts, which are both parameterized as deep neural networks. A generator G produces synthetic data given a noise variable input, while a discriminator D attempts to identify whether a sample comes from the real data

distribution X_r or the generated data distribution X_g . In other words, the discriminator is trained to estimate the probability of a given sample from the real data distribution. The generator is optimized to trick the discriminator to offer a high probability for the generated data. The two parts are optimized simultaneously to find a Nash equilibrium. More formally, the procedure can be expressed as a mini-max function:

$$\begin{aligned} \min_{\theta_G} \max_{\theta_D} L(X_r, X_g) &= \mathbb{E}_{x_r \sim X_r} [\log(D(x_r))] \\ &+ \mathbb{E}_{z \sim Z} [\log(1 - D(G(z)))] \\ &= \mathbb{E}_{x_r \sim X_r} [\log(D(x_r))] \\ &+ \mathbb{E}_{x_g \sim X_g} [\log(1 - D(x_g))], \end{aligned} \quad (9)$$

where θ_g and θ_d represent the parameters of the generator and discriminator, respectively, and Z can be a uniform noise distribution or a Gaussian noise distribution.

The function is optimized in two steps: (i) Maximize it by fixing G and X_g , and obtain the optimum of D ; (ii) Minimize the function by the optional D , and then minimize the Jensen-Shannon divergence between X_r and X_g . The game achieves equilibrium if and only if $X_r = X_g$.

Although GAN has shown great success in realistic data generation, it suffers from some major problems, such as nonconvergence, mode collapse and diminished gradient. Researchers believed that the Jensen-Shannon divergence could lead to vanishing gradients, which was the main reason for the GAN's instability. In real-world tasks such as image generation, the distribution of real images always lies in low-dimensional manifolds, and the distribution of generated images also rests in low-dimensional manifolds. The two distributions are almost certainly disjoint and have no overlaps. In this situation, the Jensen-Shannon divergence between the two distributions is a fixed number, which cannot provide useful gradients for GAN training.

Arjovsky *et al* (2017) adopted the Wasserstein distance, which is also called the earth mover's distance (EM distance), in GAN training to solve the instability problem. The distance formula for the continuous probability domain is

$$W(X_r, X_g) = \inf_{\gamma \sim \Pi(X_r, X_g)} \mathbb{E}_{(x_r, x_g) \sim \gamma} [|x_r - x_g|], \quad (10)$$

where $\Pi(X_r, X_g)$ is the set of all possible joint probability distributions between X_r and X_g . For the Wasserstein distance, even if the two distributions have no overlaps, it can still provide useful and smooth gradients for GAN training. However, it is difficult to implement the infimum of equation (10). An alternative method for calculating the Wasserstein distance

in reality is to apply its Kantorovich-Rubinstein duality:

$$\begin{aligned} W(X_r, X_g) &= \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x_r \sim X_r} [f(x_r)] \\ &- \mathbb{E}_{x_g \sim X_g} [f(x_g)], \end{aligned} \quad (11)$$

where f denotes the set of 1-Lipschitz functions and K is a constant number. In realistic implementations, f is replaced by discriminator D and $\|f\|_L \leq K$ is replaced by $\|D\|_L \leq 1$.

There are many methods for realizing the 1-Lipschitz constraint in WGAN. One possible method is to restrict the parameters of the discriminator in a limited range, such as -0.1 to 0.1. However, this weight-clipping method will introduce some problems. The model may produce poor quality data and does not converge since clipping reduces the capacity of the model. Another method is to use gradient penalty (Gulrajani *et al* 2017). In this method, an extra penalty term is added to the loss function:

$$\begin{aligned} \min_{\theta_G} \max_{\theta_D} L(X_r, X_g) &= \mathbb{E}_{x_r \sim X_r} [D(x_r)] \\ &- \mathbb{E}_{x_g \sim X_g} [D(x_g)] \\ &- \lambda \mathbb{E}_{\hat{x} \sim \hat{X}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \end{aligned} \quad (12)$$

where λ is a hyperparameter controlling the trade-off between the original objective and gradient penalty, and \hat{x} denotes the data points sampled from the straight line between the real distribution X_r and the generated distribution X_g :

$$\begin{aligned} \hat{x} &= \alpha x_r + (1 - \alpha) x_g, \\ \alpha &\sim U[0, 1], x_r \sim X_r, x_g \sim X_g. \end{aligned} \quad (13)$$

3.3. cWGAN

In this paper, we propose the cWGAN and apply it to EEG-based emotion recognition. As shown in figure 3, we can generate data with specified categories by using cWGAN. This method is based on the gradient penalty version of WGAN. The cWGAN is formulated as

$$\begin{aligned} \max_{\theta_D} L(X_r, X_g, Y_r) &= \mathbb{E}_{x_r \sim X_r, y_r \sim Y_r} [D(x_r | y_r)] \\ &- \mathbb{E}_{x_g \sim X_g, y_r \sim Y_r} [D(x_g | y_r)] \\ &- \lambda \mathbb{E}_{\hat{x} \sim \hat{X}, y_r \sim Y_r} [(\|\nabla_{\hat{x}} D(\hat{x} | y_r)\|_2 - 1)^2], \end{aligned} \quad (14)$$

$$\min_{\theta_G} L(X_g, Y_r) = -\mathbb{E}_{x_g \sim X_g, y_r \sim Y_r} [D(x_g | y_r)], \quad (15)$$

where Y_r represents the category distribution of the real data, and \hat{x} is defined in equation (13). In this

work, λ is set to 10 because we find that this value could make the training procedure more stable in our preliminary experiment. The last term in equation (14) penalizes the model if the gradient norm moves away from its target norm. The Lipschitz constraint is realized, and the model almost loses no capacity. Thus, cWGAN can generate data with high quality and converge quickly. Since the discriminator loss (D -loss) is the Wasserstein distance between the two conditional distributions can represent the training procedure for cWGAN.

Algorithm 1 The work flow of sVAE and sWGAN

Input: Real dataset $X_r = \{x_r^i\}_{i=1}^m$ and corresponding labels $Y_r = \{y_r^i\}_{i=1}^m$ and `thre_hold`
Output: Generated data $X_g = \{x_g^i\}_{i=1}^n$ and corresponding labels $Y_g = \{y_g^i\}_{i=1}^n$
 1. $X_g, Y_g = \text{Null}, \text{Null}$
 2. **repeat**
 3. $X_{all_g} = \text{sVAE}(X_r, \text{noise})$ or $X_{all_g} = \text{sWGAN}(X_r, \text{noise})$
 4. $X_{tr}, Y_{tr} = X_r \cup X_g, Y_r \cup Y_g$
 5. `model = classifier_train(X_{tr}, Y_{tr})`
 6. $Y_{all_g}, \text{class_conf} = \text{classifier_test}(\text{model}, X_{all_g})$
 7. **for** i in X_{all_g}
 8. **if** `class_conf[i] > thre_hold`
 9. $X_g, Y_g = X_g \cup X_{all_g}[i], Y_g \cup Y_{all_g}[i]$
 10. **end if**
 11. **end for**
 12. **until** `len(X_g) == n`
 13. **return** X_g, Y_g

3.4. sVAE and sWGAN

In cWGAN, we append all of the generated data to the training dataset. Here, we consider another strategy in which partially generated data are adopted to enlarge the training dataset in sVAE and sWGAN. These two methods are based on the observation that the generated data have different qualities, and only generated data with high quality are selected as new training data. This procedure has two steps: a) we generate some data by VAE or WGAN; b) we choose the generated data with high quality to enlarge the dataset. We repeat the above two steps until we obtain enough training data.

In this work, we use the classification confidence to examine data quality. We first train a classifier with the original training dataset and then use the trained classifier to classify the generated data, and only data with high classification confidence (higher than the `thre_hold`) are appended to the training dataset. We train a new classifier with the appended dataset and repeat the two steps mentioned above until we have enough generated data. We present the algorithm in Algorithm 1.

3.5. cVAE

In this method, we aim to generate data with the specified category. As shown in figure 4, to control the generated category, an extra label is added to the

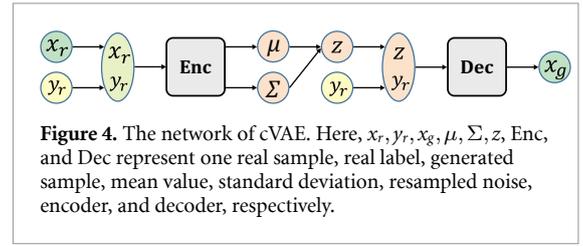


Figure 4. The network of cVAE. Here, $x_r, y_r, x_g, \mu, \Sigma, z$, Enc, and Dec represent one real sample, real label, generated sample, mean value, standard deviation, resampled noise, encoder, and decoder, respectively.

encoder and decoder. We first feed the training data point and the corresponding label to the encoder, then we concatenate the hidden representation with the corresponding label and feed it to the decoder to train the network. Then, we can generate data with the specified label by feeding the decoder with the noise sampled from the Gaussian distribution and the assigned label after the training process. Therefore, the cVAE (Kingma and Welling 2014) can be formulated as,

$$\max_{\lambda, \phi} ELBO = \sum_i ELBO_i = \sum_i \sum_k [(x_i - \hat{x}_i)^2 + \frac{1}{2}(\Sigma(x_i|y_i) + \mu^2(x_i|y_i) - 1 - \log \Sigma(x_i|y_i))]. \quad (16)$$

3.6. Gaussian noise

One of the straightforward augmentation methods is adding Gaussian noise (Gau) to the original training data, whose probability density function obeys a Gaussian distribution:

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \quad (17)$$

where z is a random variable, μ means expectation and σ is the standard deviation. In our experiment, μ is set to 0 and σ is set to 0.001. Intuitively, more training data can be generated while preserving the characteristics of the original data by adding Gaussian noise.

3.7. Rotational data augmentation

Rotational data augmentation (RDA) was proposed by Krell and Kim (2017), which aims to create data with strong spatial robustness, since there might be spatial shifts of EEG caps within sessions and between sessions during the experiments. To address this problem, RDA generates artificial data associated with the electrodes' new positions by adding rotations on three coordinates. According to their result (Krell and Kim 2017), augmentation around the y -axis and z -axis increased the performance, especially around the z -axis. Therefore, we choose to perform the rotations around the z -axis. Specifically, we set an angle between 12° and 24° over all subjects and calculate the new data by interpolation based on radial basis functions.

3.8. Classifier

In this paper, we implement two kinds of classifiers: SVMs and deep neural networks.

The deep neural network is a neural network with multiple hidden layers. Here, we randomly add some residual functions between two layers. The idea of the residual function is borrowed from Resnet (He *et al* 2016). The residual function is a way to avoid the problem of vanishing gradients, and it does this by using shortcuts to jump over some layers. Because the numbers of different nodes are different, the dimensions of input and output are different. Therefore, we can use a linear projection to match the dimensions. This function can be expressed as follows:

$$y = F(x) + W_s x, \quad (18)$$

where x and y are the input and output vectors, respectively, $F(x) = W_2 \sigma(W_1 x)$, which means two fully connected layers and a ReLU function, σ , between them, and W_s is the linear projection to change the dimension. The output should go through another ReLU function before it is passed to the next layer.

4. Experimental settings

In this section, we describe the details about the two datasets, data preprocessing, performance evaluation, and hyperparameters. For reproducing the results of this paper and enhancing the cooperation in related research fields, the datasets and codes used in this study will be freely available to the academic community.

4.1. Dataset description

The SEED dataset (Zheng and Lu 2015) contains the EEG signals of 15 participants. They were required to watch 15 well-prepared video clips that can elicit exactly one of the three kinds of emotion: positive, neutral, and negative. The criteria of film clip selection ensure that Each clip is well-edited to create coherent emotion eliciting and maximizing emotional meanings. In addition, each clip can explicitly elicit one exact kind of emotion, and the time of the clips is enough but not too long to elicit the participants' corresponding emotion sufficiently. The order of presentation is arranged so that two film clips targeting the same emotion are not shown consecutively. Each participant took part in the experiment three times with an interval of at least 7 days. The signals were sampled at a rate of 1 000 Hz with an ESI NeuroScan System from a 62-electrode headset.

The DEAP dataset (Koelstra *et al* 2012) contains the EEG and peripheral physiological signals of 32 participants as they watched 40 one-minute-long excerpts of music videos. The music videos were selected from 120 one-minute extracts of music videos rated from an online self-assessment by 14–16

volunteers based on valence, dominance, arousal, like, and familiarity. Valence, arousal, dominance and like were rated directly after each trial on a continuous 9-point scale using a standard mouse self-assessment. The signals were sampled at 512 Hz with 48 channels. (32 EEG channels, 12 peripheral physiological channels including galvanic skin response and temperature, 3 unused channels and 1 status channel). The signals from EEG channels are sampled according to an international 10–20 system.

4.2. Data preprocessing

Previous works have shown that the DE feature of EEG signals is efficient for EEG-based emotion recognition (Zheng and Lu 2015, Zheng *et al* 2019, Yang *et al* 2018). Therefore, we generate DE features to augment the datasets. We also generate the PSD feature, which is a conventional feature for EEG-based emotion recognition, to verify our method. Since both the SEED and DEAP datasets have been preprocessed, we use the short-time Fourier transform (STFT) with a 1-s-long nonoverlapping overlapping Hanning window to extract the PSD feature of the EEG signal from the two datasets directly. For the Gaussian distribution, the DE feature is defined as

$$h(X) = - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = \frac{1}{2} \log 2\pi e\sigma^2, \quad (19)$$

where X represents the Gaussian distribution $N(\mu, \sigma^2)$, and π and e are constants. Shi *et al* (2013) demonstrated that the value of DE is equal to the logarithmic spectral energy for a fixed-length EEG sequence in a certain band. According to their result, we extracted the DE feature from the preprocessed EEG signal of the two datasets.

Considering the dynamic characteristics of EEG-based emotion recognition tasks, we employ the linear dynamic system approach to filter the PSD and DE features, which has also been adopted in previous works (Zheng and Lu 2015, Zheng *et al* 2019).

PSD and DE features are extracted from five frequency bands: δ : 1–3 Hz, θ : 4–7 Hz, α : 8–13 Hz, β : 14–30 Hz, and γ : 31–50 Hz for the SEED dataset (Zheng and Lu 2015). Therefore, both of these features have 310 dimensions (62 channels \times 5 frequency bands). For each experiment, there were 3 394 labeled samples. In this work, we viewed the SEED dataset as a three-category classification problem.

We also extracted PSD and DE features for the DEAP dataset. Since the δ band was filtered in this dataset, we only computed the two features of four frequency bands: θ , α , β , and γ . In this time, both features had 128 dimensions (32 channels \times 4 frequency bands). Each experiment had 2,400 labeled samples. Here, we adopted a four-category emotion model using valence and arousal values: high valence

(level > 5) and high arousal (level > 5), high valence (level > 5) and low arousal (level ≤ 5), low valence (level ≤ 5) and high arousal (level > 5), and low valence (level > 5) and high arousal (level > 5).

4.3. Evaluation details

We adopted 5-fold cross-validation, which has been applied in the existing study (Zheng *et al* 2019), for each experiment on the two datasets. For each time, 4 folds of the dataset were selected as the original training data and 1 fold was selected as the test data. The division of each dataset was the same as Zheng *et al*'s work.

Moreover, we added different numbers of the artificial sample generated by different data augmentation methods to each original training dataset. And the test data which only contained the real data was the same for each data augmentation experiment. In a word, we trained 5 recognition models and computed the average recognition accuracy of the five models as the recognition accuracy of each data augmentation experiment. Each model had the same hyperparameters.

For a specific generated number and data augmentation method, we regarded the average accuracy of all data augmentation experiments as the final accuracy and the standard deviations of these experiments as the final standard deviation. For the SEED dataset, there were 45 experiments and nearly 678 samples for each fold. For the DEAP dataset, there were 32 experiments and 480 samples for each fold.

4.4. Hyperparameter details

The hyperparameter settings of Gau and RDA methods have been mentioned above.

For cVAE method, to optimize the network structure, we performed a random search on the number of network layers. The number of layers was searched from 2 to 4 for both encoder and decoder. The number of hidden nodes for each layer was randomly searched. The dimensions of auxiliary labels were 3 for SEED and 4 for DEAP. For sVAE method, we had a similar search strategy as cVAE.

For cWGAN method, we also performed a random search on the number of network layers. The number of layers was searched from 3 to 5 for both generator and discriminator. The number of hidden nodes for each layer was randomly searched. And the dimensions of auxiliary labels were 3 for SEED and 4 for DEAP. For sWGAN method, we had a similar search strategy as cWGAN.

For sVAE and sWGAN methods, the *threshhold* in Algorithm 1 was randomly searched from 0.4 to 0.99.

In the SVM classifier, we used the linear kernel. The parameter *c* was searched from 2^{-10} to 2^{10} to find the optimal value.

We performed a random search on the number of network layers and size of batches of classifier of deep

neural network. The number of layers was searched from 4 to 8. The size of batches was randomly selected from 128, 256 and 512. Both networks with residual functions and without residual functions were searched. For the network with residual functions, the residual functions were applied every two layers. The input dimension was determined by the corresponding input feature, and the dimension of the output label was 3 for the SEED dataset and 4 for the DEAP dataset. The number of hidden nodes for each layer was randomly searched. The ReLU activation function was used for all hidden layers. We normalize PSD and DE features before feeding them to the networks.

In addition, we applied Adam optimizer to optimize all the networks mentioned above. And the initial learning rate was randomly selected from 0.00001 to 0.001.

5. Experiments and results

In this section, we first perform a systematic experimental study to evaluate the effectiveness and generalization ability of our methods. We augmented different EEG-based emotion datasets by different features generated by our methods. We apply different classifiers to evaluate the performances of these generative methods. We also compare our proposed methods with conventional generative methods. Then, we visualized the generated data to show why our proposed methods work. Finally, we discuss the characteristics of the proposed methods and future work.

5.1. Different number of appended training data

We first conducted data augmentation experiments on the SEED dataset and use the SVM as the classifier. Each experiment had 3,394 samples. We generated 0, 200, 500, 1000, 3000, 5000, 10 000, 15 000 and 20 000 artificial samples of the two features and added them to the original training datasets. Here, '0' indicates that we only use the original training dataset without data augmentation. We did not generate more samples because we found that most of the experiments reach their peaks before 20 000 samples were appended. The remaining experiments reached their peaks when 20 000 samples were appended. And the *p* - values between sWGAN method and the conventional methods are all less than 0.01.

We compared the performances of different data augmentation methods when applying the PSD feature, as shown in table 1. The average accuracy was 60.3% when we only used the original training set. For conventional methods, cVAE reached its best mean accuracy of 63.4% when 3 000 samples were appended. Gau reached its optimal performance of 63.1% when adding 10 000 samples into the original training set. RDA had the best performance of 63.2% when 500 samples were appended. For

Table 1. Mean accuracies/standard deviations of SVMs on the SEED dataset and appending datasets using PSD feature generated by different methods. ‘↑’ represents the maximize accuracy improvement of different methods and has the same meaning in tables 2, 3, and 4.

methods	No. of append									
	0	200	500	1000	3000	5000	10 000	15 000	20 000	↑
cVAE + SVM	60.3/15.9	62.7/15.7	62.8/15.4	63.4/14.6	63.4/14.8	63.3/ 14.3	62.5/14.8	61.8/14.8	61.6/14.5	3.1
Gau + SVM	60.3/15.9	61.4/15.6	61.7/15.5	61.7/15.7	62.5/15.4	62.5/15.6	63.1/15.0	62.7/15.5	62.8/15.4	2.8
RDA + SVM	60.3/15.9	62.6/15.7	63.2/15.4	62.9/15.8	62.0/15.6	62.1/ 15.2	61.5/15.8	61.9/15.9	61.1/16.3	2.9
cWGAN + SVM	60.3/15.9	62.7/15.5	63.6/15.6	63.5/15.6	64.0/15.6	64.4/15.5	65.0/15.6	65.2/15.5	64.9/15.5	4.9
sVAE + SVM	60.3/ 15.9	62.7/16.9	62.6/16.6	63.3/16.6	62.8/16.9	63.1/16.5	63.4/17.4	63.5/17.2	63.2/17.4	3.2
sWGAN + SVM	60.3/15.9	65.2/ 14.5	66.0/14.8	66.8/14.9	67.0/14.7	67.0/14.7	67.4/14.8	67.3/15.2	67.7/15.1	7.4

Table 2. Mean accuracies/standard deviations of SVMs and deep neural network (DNNs) with shortcut layers on the SEED dataset and appending datasets using DE feature generated by different methods.

methods	No. of append									
	0	200	500	1000	3000	5000	10 000	15 000	20 000	↑
cVAE + SVM	84.3/8.7	84.8/8.7	85.2/8.6	85.2/8.6	84.9/8.5	84.9/ 8.5	84.5/8.9	84.0/8.9	84.0/8.9	0.9
cVAE + DNN	83.3/8.2	83.9/9.3	84.9/8.5	86.1/8.0	86.5/7.5	86.1/8.2	85.1/7.8	85.1/8.5	86.1/8.2	3.2
Gau + SVM	84.3/8.7	84.6/8.7	84.8/8.6	84.9/8.6	85.1/8.5	85.0/8.6	85.0/8.7	84.8/8.6	84.8/8.5	0.8
Gau + DNN	83.3/8.2	85.9/7.6	85.7/7.9	85.0/8.5	85.6/7.3	85.3/8.3	86.2/8.2	84.9/8.5	85.8/7.8	2.9
RDA + SVM	84.3/8.7	85.4/9.0	85.5/9.1	85.5/9.0	85.4/8.9	85.6/8.8	84.7/9.1	84.3/9.3	84.3/9.3	1.3
RDA + DNN	83.3/8.2	85.7/9.9	83.4/9.5	82.1/9.7	78.2/9.6	77.6/10.9	77.6/10.1	74.6/8.9	75.7/9.1	2.4
cWGAN + SVM	84.3/8.7	87.0/8.6	87.2/8.5	86.8/8.4	87.0/8.4	87.0/8.5	87.4/8.0	87.4/7.9	87.1/ 7.9	3.1
cWGAN + DNN	83.3/8.2	86.6/7.7	89.2/7.9	89.7/8.3	91.6/6.7	90.9/7.9	90.6/7.9	90.6/8.8	90.7/7.8	8.3
sVAE + SVM	84.3/8.7	87.4/7.9	87.5/7.6	87.8/7.6	86.8/8.1	86.1/8.6	85.2/8.7	84.7/8.1	84.5/8.1	3.5
sVAE + DNN	83.3/8.2	85.8/8.8	86.8/7.3	87.5/8.6	87.2/6.8	84.1/6.7	84.0/6.5	82.2/ 6.2	80.4/6.6	4.2
sWGAN + SVM	84.3/8.7	87.9/8.4	88.9/8.3	89.7/7.9	90.1/7.6	90.7/7.8	90.8/7.7	90.8/7.3	90.8/7.4	6.5
sWGAN + DNN	83.3/8.2	91.4/7.2	91.5/6.4	93.5/5.7	93.5/5.8	93.0/5.8	93.1/ 5.6	91.7/6.0	92.2/5.7	10.2

Table 3. Mean accuracies/standard deviations of SVMs on the DEAP dataset and appending datasets using PSD feature generated by different methods.

methods	No. of append									
	0	200	500	1000	3000	5000	10 000	15 000	20 000	↑
cVAE + SVM	42.7/9.6	43.7/9.5	44.5/ 8.7	44.2/9.3	44.9/8.8	44.6/8.8	44.1/8.9	44.1/9.1	43.4/9.1	2.2
Gau + SVM	42.7/9.6	43.2/9.5	43.6/ 9.2	43.7/9.8	43.9/9.7	44.5/9.3	44.0/9.6	43.9/9.6	43.9/9.5	1.8
RDA + SVM	42.7/9.6	42.8/10.0	43.0/9.7	44.1/9.2	44.3/9.2	44.9/8.5	44.7/8.9	44.9/9.0	45.2/8.9	2.5
cWGAN + SVM	42.7/9.6	44.1/9.6	44.2/9.7	44.8/9.1	44.9/8.8	45.0/8.9	44.9/ 8.7	44.9/9.0	44.8/9.2	2.3
sVAE + SVM	42.7/9.6	44.7/8.4	45.1/ 8.1	45.6/8.5	45.6/8.3	45.8/8.3	46.1/8.2	46.1/8.5	45.9/8.7	3.4
sWGAN + SVM	42.7/ 9.6	45.8/10.6	45.8/11.0	46.4/10.4	46.7/10.3	46.9/10.4	47.1/10.1	47.4/10.0	47.6/9.9	4.9

Table 4. Mean accuracies/standard deviations of SVMs and deep neural networks with shortcut layers on the DEAP dataset and appending datasets using DE feature generated by different methods.

methods	No. of append									
	0	200	500	1000	3000	5000	10 000	15 000	20 000	↑
cVAE + SVM	45.4/8.2	46.3/8.1	46.8/8.0	47.2/7.8	47.9/7.7	47.8/7.7	48.1/7.6	48.0/7.7	47.8/7.6	2.7
cVAE + DNN	44.9/4.0	46.6/4.4	45.8/3.8	45.5/4.9	45.7/4.6	46.5/4.9	45.7/3.9	45.9/ 3.6	45.9/4.5	1.7
Gau + SVM	45.4/8.2	46.1/ 8.0	46.0/8.2	46.1/8.1	46.1/8.1	45.9/8.2	46.0/8.2	45.8/8.1	45.8/8.2	0.7
Gau + DNN	44.9/4.0	45.9/ 3.7	46.9/4.2	45.5/4.1	45.5/4.5	46.2/4.6	45.8/4.5	45.6/4.6	46.2/4.5	2.0
RDA + SVM	45.4/8.2	45.9/8.2	45.9/8.2	46.1/8.5	46.3/8.1	46.3/8.1	46.1/7.8	46.0/7.8	45.9/7.9	0.9
RDA+DNN	44.9/4.0	46.3/4.5	46.8/4.3	46.1/3.7	46.1/4.7	46.0/4.1	45.8/ 3.4	45.1/3.7	45.9/4.1	1.9
cWGAN + SVM	45.4/ 8.2	47.3/8.2	47.9/8.2	48.0/8.3	48.8/8.3	48.9/8.7	48.5/8.4	48.2/8.8	48.0/8.9	3.5
cWGAN + DNN	44.9/4.0	45.4/4.4	45.9/ 4.0	47.2/5.1	47.0/4.4	46.9/4.8	47.1/4.6	47.5/4.5	46.9/4.8	2.6
sVAE + SVM	45.4/8.2	47.6/7.1	48.3/7.1	48.2/7.3	48.3/6.8	48.4/6.8	48.1/7.1	48.2/7.0	48.1/7.1	3.0
sVAE + DNN	44.9/4.0	47.3/ 3.9	47.7/4.6	47.6/4.2	47.3/4.2	49.3/5.0	47.7/5.3	47.7/4.4	47.6/6.0	4.4
sWGAN + SVM	45.4/8.2	47.6/7.7	47.9/7.5	48.7/7.4	49.6/7.2	49.9/6.8	50.3/7.0	50.8/6.9	50.4/ 6.7	5.4
sWGAN + DNN	44.9/ 4.0	47.2/4.1	47.7/4.7	47.6/4.4	48.2/4.9	49.1/5.6	47.5/4.6	48.5/5.2	47.6/5.3	4.2

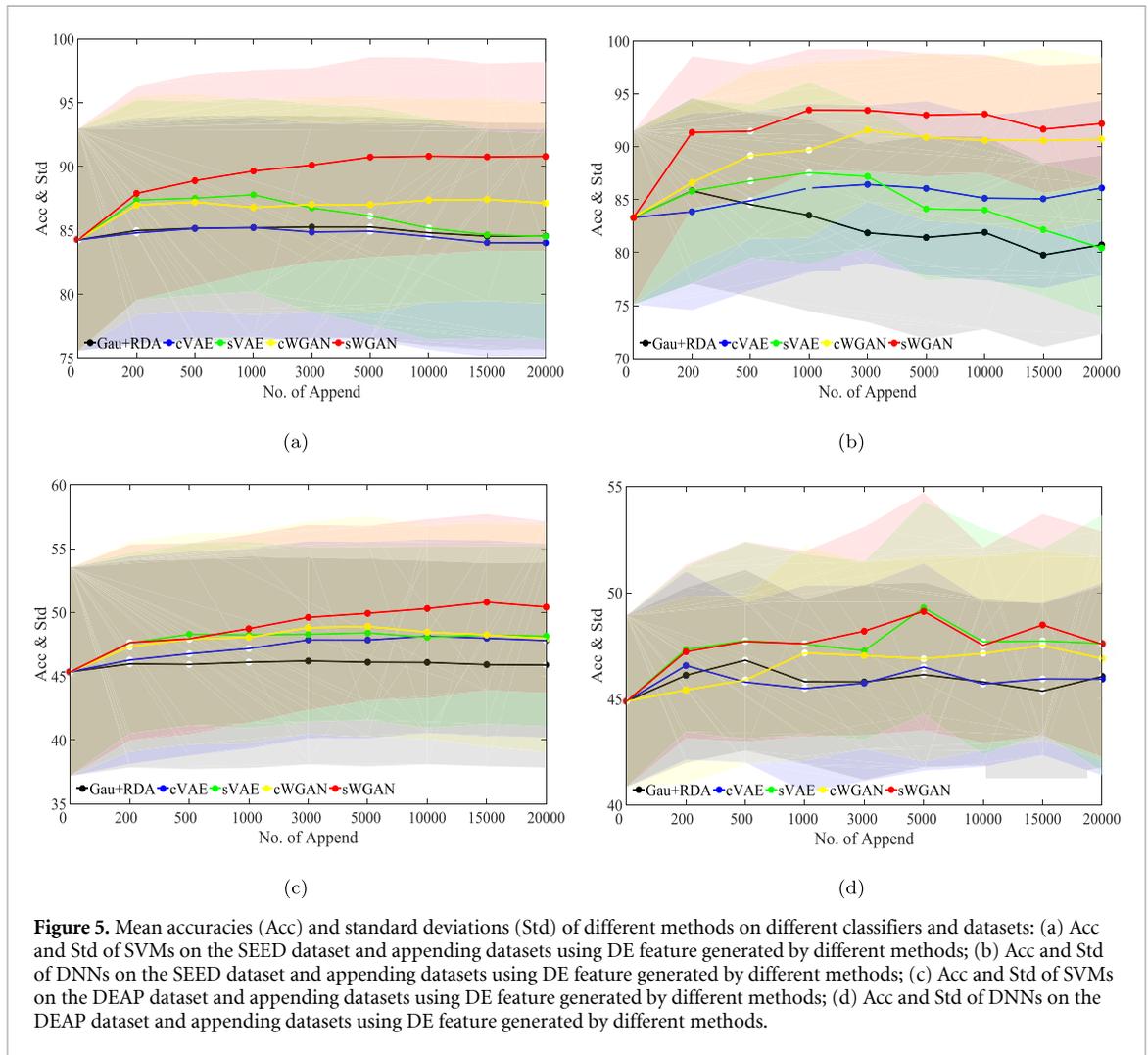


Figure 5. Mean accuracies (Acc) and standard deviations (Std) of different methods on different classifiers and datasets: (a) Acc and Std of SVMs on the SEED dataset and appending datasets using DE feature generated by different methods; (b) Acc and Std of DNNs on the SEED dataset and appending datasets using DE feature generated by different methods; (c) Acc and Std of SVMs on the DEAP dataset and appending datasets using DE feature generated by different methods; (d) Acc and Std of DNNs on the DEAP dataset and appending datasets using DE feature generated by different methods.

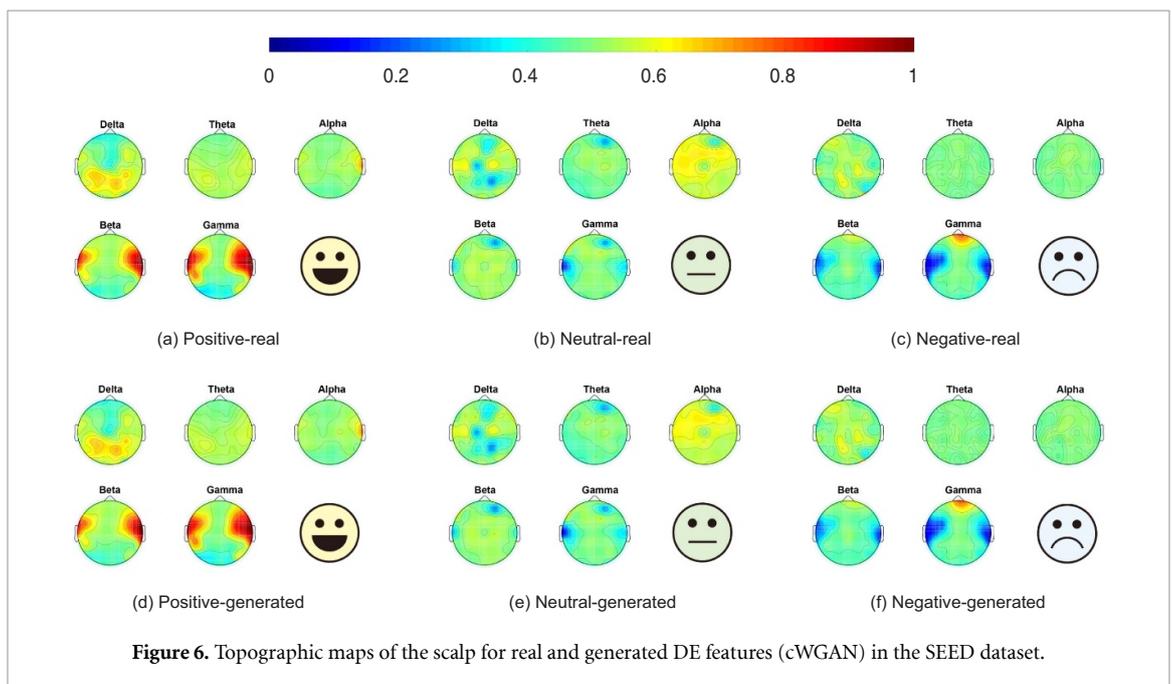


Figure 6. Topographic maps of the scalp for real and generated DE features (cWGAN) in the SEED dataset.

our methods, cWGAN achieved its best mean accuracy of 65.2% when 15 000 samples were appended. When 15 000 samples were appended, sVAE

reached its best mean accuracy of 63.5%. sWGAN achieved its best mean accuracy of 67.7% when 20 000 samples were appended. According to table 1, our

methods achieved better performance than conventional methods. sWGAN had the best performance among all the methods.

Table 2 illustrates the results of the data augmentation methods for the DE feature. For SVM, the baseline was 84.3%. For conventional data augmentation methods, cVAE reached its best result of 85.2% when 1000 sampled data points were appended. The best accuracy for Gau was 85.1% with 3000 augmented data points. The best mean accuracy of RDA was 85.6% when 5000 samples were appended. For our methods, cWGAN reached its best mean accuracy of 87.4% when appending 15 000 samples. For the two selective augmentation methods, sVAE achieved the best mean accuracy of 87.8% when 1000 samples were appended, and sWGAN achieved the best mean accuracy of 90.8% when 10 000 samples were appended.

5.2. Classification with deep neural networks

To increase the reliability of the performance comparison of different data augmentation approaches, we also implemented deep neural networks with shortcut layers to build the affective models. Considering that the DE feature is better for the PSD feature in emotion recognition tasks and that the PSD feature had similar improvements in terms of the mean accuracy with the DE feature, we only augmented the training data with the DE features when using the DNN as the classifier. The baseline was 83.3%. For conventional methods, cVAE, Gau, and RDA reached the best mean accuracy of 86.5% (3000 samples), 86.2% (10 000 samples), and 85.7% (200 samples), respectively. For our proposed methods, the best mean accuracy of cWGAN is 91.6% when we added 3000 samples. The two selective methods obtained the best mean accuracy of 87.5% and 93.5% when we added 1000 samples, respectively. The results in table 2 demonstrate that our methods had better performance than conventional methods. The sWGAN achieved the best performance for both classifiers.

5.3. Generated data with two different features

For the DEAP dataset, we also used different data augmentation methods to augment PSD and DE features. Each experiment had 2400 samples. We generated the same number of samples as mentioned above.

Table 3 shows the mean accuracies and standard deviations of PSD data augmentation. The mean accuracy of the 4-category emotion recognition model was 42.7% when we only used the original training data. For conventional methods, cVAE reached the best mean accuracy of 44.9% when 3000 samples were appended. The best performance for Gau was 44.5% when 5000 samples were appended. RDA reached the best mean accuracy of 45.2% when 20,000 samples were appended. For our methods, cWGAN obtained the best mean accuracy of 45.0%

when we added 5000 generated samples to the original training dataset. sVAE had the best mean accuracy of 46.1% when 15 000 samples were generated. sWGAN achieved its best mean accuracy of 47.6% when 20 000 samples were appended. Our methods also showed better performance, and sWGAN had the best performance in terms of accuracy.

Table 4 presents the results of DE data augmentation. For SVM, the baseline was 45.4%. For conventional methods, cVAE had the best mean accuracy of 48.1% when 10 000 samples were appended. The best accuracy for Gau was 46.1% when we appended 1000 samples. RDA obtained the best mean accuracy of 46.3% when the number of appended samples was 3000. For our methods, cWGAN obtained the best mean accuracy of 48.9% when 5000 samples were appended. sVAE reached the best mean accuracy of 48.4% when 5000 samples were appended, and sWGAN obtained the best mean accuracy of 50.8% when 15 000 samples were appended.

For DNN, the classification accuracy was 44.9% when no data augmentation method was applied. For conventional methods, cVAE, Gau, and RDA reached the best mean accuracies of 46.6% (200 samples), 46.9% (500 samples), and 46.8% (500 samples), respectively. For our methods, cWGAN, sVAE, and sWGAN achieved the best mean accuracy of 47.5% (15 000 samples), 49.3% (5000 samples), and 49.1% (5000 samples). We also observed that our methods showed better performance than conventional methods. sWGAN had the best mean accuracy when applying SVM as the classifier, while sVAE had the best performance when applying DNN as the classifier.

As we can see from the above results, the DE feature had better mean accuracies than the PSD feature on both datasets, and the standard deviations were smaller. These results were consistent with previous studies (Zheng and Lu 2015, Zheng *et al* 2019). In addition, compared with conventional methods, our methods were more efficient for improving the performance of emotion recognition models. For the SEED dataset, the mean accuracy improved 10.2% with DE features when we used sWGAN as the data augmentation method and adopted DNN as the classifier. The DEAP dataset had the highest improvement of 5.4% in terms of mean accuracy when sWGAN was adopted as the data augmentation method and SVM was used as the classifier. Moreover, the data augmentation methods were more efficient for DNN in most cases.

In addition, we observe that all the data augmentation methods (both ours and conventional methods) reached their peaks, and then their performance decayed when we gradually increased the number of appended samples. However, for our methods, most of the experiments still showed better performance compared with their baselines when appending fewer

than 20,000 generated samples. Although experiments with different datasets, features, classifiers, and methods reached different peaks, our results show that the peaks appeared before the training datasets were enlarged 10 times.

As shown in figure 5, we plotted the mean accuracies and standard deviations of different methods on different classifiers and datasets. We only shown the results of DE feature because DE feature had better performance than PSD feature and the two features had the similar tendency in terms of mean accuracy. And the results of Gau and RDA were averaged. Compared with the conventional methods, our methods had better mean accuracies in the most experiments. Besides, GAN-based methods (cWGAN and sWGAN) shown better performance than VAE-based (cVAE and sVAE) methods in most cases. Moreover, the selective methods (sVAE and sWGAN) were better than conditional methods (cVAE and cWGAN) in most of the experiments. Specially, sWGAN always had better mean accuracies than cWGAN.

5.4. Visualization of the generated data

We visualize the generated data with two methods, two-dimensional circular view of the scalp and two-dimensional visualization using t-SNE, to show why our proposed methods work. We selected cWGAN as the generated method and the SEED dataset (DE feature) to represent our results since sVAE and sWGAN have similar vision performance.

Figure 6 depicts the two-dimensional circular view of the scalp. The generated data have a similar data distribution as the real data. For positive emotion, the lateral areas of both real and generated data are more activated in beta and gamma bands than the other two emotions. For neutral emotion, both the real and generated data had high alpha responses. For negative emotion, high gamma responses at prefrontal sites appeared in real and generated data. These phenomena indicated that our methods can capture the information of the real data distribution. Therefore, the generated samples can be appended to the training set to enhance the performance of the affective models.

As shown in figure 7, we plotted the distributions of real and generated DE features (generated by cWGAN) by t-SNE (Maaten and Hinton 2008). Data from each emotion was clustered in the latent space, and the generated data were close to the corresponding real data, which implies that the generated data carry enough realistic information. This phenomenon also indicates that the data generated by our methods can be used to augment the training dataset.

In addition, the distribution of real data was sparse, and the boundaries of different categories in the data manifold were not obvious. The generated data supplemented the training data manifold, which led to better margins for the classifier. Therefore, we

can improve the classification performance by training the classifier with the generated data. We can also explain this phenomenon from another point of view. The generated data have a similar data distribution to the real data, but they are not the real data. Therefore, the generated data not only carry realistic information but also have diverse information. The classifier trained by the augmented data was more robust. This phenomenon is also consistent with the aforementioned classification results.

However, the possibility of generating bad quality samples increased when we added the generating number. This phenomenon occurs no matter what generative methods we apply. For example, we wanted to generate a sample of positive emotions in the SEED dataset, but we might obtain a sample that is more similar to a negative sample by the generative model. We called this sample a bad quality sample. In figure 7, we can find some bad quality samples. For example, some generated neutral samples (red points) were more close to the real positive samples (blue lines). In this case, the bad quality sample misled the classifier, and the classification accuracy decreased. We can also find a similar phenomenon in the above tables: the accuracies decayed when too many generated data were appended.

5.5. Discussions on different affective models

The abovementioned results show that the performance of the emotion recognition models can be improved by using our proposed data augmentation methods. We achieved performance improvements in different datasets, features, and classifiers, which demonstrates the generalization ability and effectiveness of our methods. Although all three proposed methods improved the performance of EEG-based emotion recognition tasks, they had some differences in terms of stability, accuracy and time usage.

For stability, sVAE had better performance than cWGAN and sWGAN. Although the WGAN had good convergence performance and was more stable than the original GAN, it may collapse because of adversarial training. However, VAE is more stable.

For accuracy, sWGAN had better classification performances than sVAE most of the time. This phenomenon indicates that GAN can capture more latent information than VAE. Therefore, the data generated by GAN are more useful for building the recognition model than those generated by VAE. In addition, sWGAN always performed better than cWGAN on both datasets, which indicates that the selective methods are more efficient at improving emotion recognition models.

For time usage, cWGAN had a quicker convergence speed than sWGAN and sVAE. cWGAN uses all of the generated data without considering their quality, while sWGAN and sVAE need to select the generated data and use the high-quality data to augment

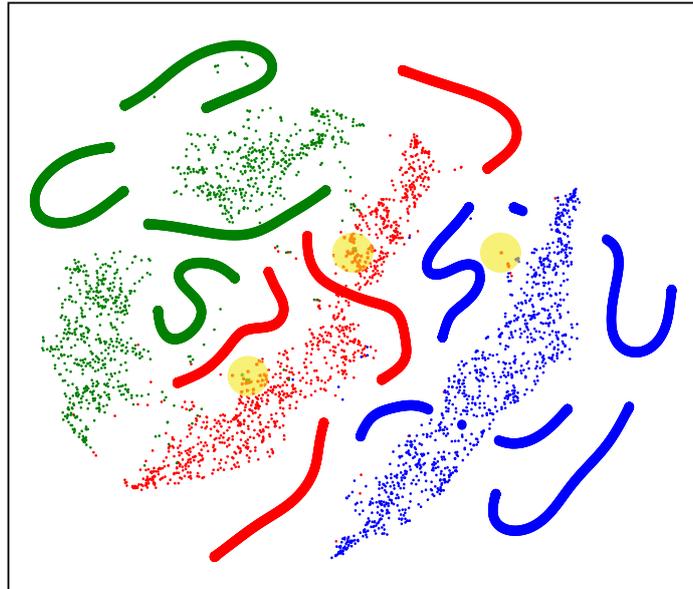


Figure 7. Two-dimensional visualizations of the real and generated DE feature (cWGAN) of one subject in the SEED dataset. Data points with red, green and blue colors represent three emotions of negative, neutral and positive, respectively. The lines represent the real data, thin points represent the generated data, and the yellow circles denote bad quality samples generated. Note that we use solid circle to represent real data, but it looks like lines more than circles because the real data are very dense.

the training set. Therefore, the two methods require more computation time to determine the quality of the generated data.

5.6. Future work

Although the experimental results of this work demonstrate that our proposed data augmentation methods are promising ways to enhancing the performance of emotion recognition models, there is still room for improvement in the following aspects. a) We did not consider the temporal dependency when generating the EEG features. In future work, we will study the data augmentation methods for generating temporal raw EEG data. b) Although we have found the affective models could achieve the best performance when the number of the added generated data is less than ten times of the original training dataset, how many generated samples should be added to the original training dataset to achieve the best affective models is still an open question. c) Recently, various studies indicated that multimodal methods could achieve appealing emotion recognition results (Zheng, Liu, Lu, Lu and Cichocki 2019, Zhao *et al* 2019). In our previous work, we generated multimodal feature for enhancing EEG-based emotion recognition and obtained 4.6% and 8.9% improvements of mean accuracies on classifying three and five emotions, respectively (Luo *et al* 2019). Due to the space limitation, we did not discuss the problem of generating multimodal feature in this paper. In

future work, we will further study the data augmentation methods for multimodal emotion recognition.

6. Conclusions

In this paper, we proposed three deep generative methods for enhancing EEG-based emotion recognition by generating training data. We generated realistic-like PSD and DE features of EEG data with our proposed methods: cWAGN, sVAE, and sWGAN. We augmented the original training dataset using the generated data to improve the accuracy of EEG-based emotion recognition models. The experimental results on two emotion datasets demonstrate the effectiveness of our methods. The emotion recognition models trained on the augmented training datasets achieved 10.2% and 5.4% improvements on the SEED dataset and the DEAP dataset, respectively. By visualizing the generated data, we explained the reason for the accuracy improvements. We also studied the performance of the classifiers when adding different numbers of generated data to the original training set. We observed that the classification accuracy decayed when too many generated data were appended. Our experimental results indicate that the number of generated data should be less than ten times of the original training dataset, and then the affective models achieved the best performance. In addition, we carried out a systematic comparison between the proposed methods. We find that sWGAN had the best

performance in terms of accuracy, while it cost more time than cWGAN.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (2017YFB1002501), the National Natural Science Foundation of China (61673266 and 61976135), SJTU Trans-med Awards Research (WF540162605), the Fundamental Research Funds for the Central Universities, and the 111 Project.

ORCID iDs

Yun Luo  <https://orcid.org/0000-0001-7780-4929>

Bao-Liang Lu  <https://orcid.org/0000-0001-8359-0058>

References

- Alarcao S M and Fonseca M J 2017 Emotions recognition using EEG signals: A survey *IEEE Trans. Affective Computing* **10** 374–93
- Arjovsky M, Chintala S and Bottou L 2017 Wasserstein GAN arXiv:1701.07875
- Berthelot D, Schumm T and Metz L 2017 BEGAN: Boundary equilibrium generative adversarial networks arXiv:1703.10717
- Bocharov A V, Knyazev G G and Savostyanov A N 2017 Depression and implicit emotion processing: An EEG study *Neurophysiol. Clinique* **47** S0987705316303513
- Choi E, Biswal S, Malin B, Duke J, Stewart W F and Sun J 2017 Generating multi-label discrete electronic health records using generative adversarial networks arXiv:1703.06490
- Craik A, He Y and Contreras-Vidal J L 2019 Deep learning for electroencephalogram (EEG) classification tasks: a review *J. Neural Eng.* **16** 031001
- Dahl G E, Yu D, Deng L and Acero A 2011 Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition *IEEE Trans. Audio Speech Language Process.* **20** 30–42
- Duan R-N, Zhu J-Y and Lu B-L 2013 Differential entropy feature for EEG-based emotion classification *Int. IEEE/ Conf. on Neural Engineering (NER)* IEEE pp 81–4
- Fourati R, Ammar B, Sanchez-Medina J and Alimi A M 2020 Unsupervised learning in reservoir computing for eeg-based emotion recognition *IEEE Trans. Affective Computing* **1–1**
- Garber-Barron M and Si M 2012 Using body movement and posture for emotion detection in non-acted scenarios *IEEE Int. Conf. on Fuzzy Systems* pp 1–8
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Neural Information Process. Syst.* 2672–80
- Gregor K, Danihelka I, Graves A, Rezende D J and Wierstra D 2015 Draw: A recurrent neural network for image generation arXiv:1502.04623
- Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V and Courville A C 2017 Improved training of Wasserstein GANs *Neural Information Processing Systems* pp 5769–79
- Halevy A, Norvig P and Pereira F 2009 The unreasonable effectiveness of data *IEEE Intelligent Syst.* **24** 8–12
- Hartmann K G, Schirmer R T and Ball T 2018 EEG-GAN: Generative adversarial networks for electroencephalographic (EEG) brain signals arXiv:1806.01875
- He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. of the Conf. on Computer Vision and Pattern Recognition* pp 770–8
- Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504–7
- Jenke R, Peer A and Buss M 2014 Feature extraction and selection for emotion recognition from EEG *IEEE Trans. Affective Computing* **5** 327–39
- Katsigiannis S and Ramzan N 2018 Dreamer: A database for emotion recognition through eeg and ecg signals from wireless low-cost off-the-shelf devices *IEEE J. Biomedical Health Informatics* **22** 98–107
- Kingma D P and Welling M 2014 Auto-encoding variational bayes *Int. Conf. on Learning Representations*
- Koelstra S, Mühl C, Soleymani M, Lee J-S, Yazdani A, Ebrahimi T, Pun T, Nijholt A and Patras I 2012 DEAP: A database for emotion analysis; using physiological signals *IEEE Trans. Affective Computing* **3** 18–31
- Krell M M and Kim S K 2017 Rotational data augmentation for electroencephalographic data *2017 39th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)* pp 471–4
- Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* pp 1097–105
- Kulkarni T D, Whitney W F, Kohli P and Tenenbaum J 2015 Deep convolutional inverse graphics network *Neural Information Process. Syst.* pp 2539–47
- Ledig C et al 2017 Photo-Realistic single image super-resolution using a generative adversarial network *Computer Vision Pattern Recognition* pp 105–14
- Li J, Monroe W, Shi T, Jean S, Ritter A and Dan J 2017 Adversarial learning for neural dialogue generation arXiv:1701.06547
- Liu Z, Min W, Cao W, Chen L, Xu J, Zhang R, Zhou M and Mao J 2017 A facial expression emotion recognition based human-robot interaction system *IEEE/CAA J. Automatica Sinica* **4** 668–76
- Lotte F 2015 Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain-computer interfaces *Proc. IEEE* **103** 871–90
- Luo Y and Lu B-L 2018 EEG data augmentation for emotion recognition using a conditional Wasserstein GAN *2018 40th Annual Int. Conf. of the IEEE Eng. Med. Biol. Soc.* pp 2535–8
- Luo Y, Zhu L-Z and Lu B-L 2019 A GAN-based data augmentation method for multimodal emotion recognition *Advances in Neural Networks – Isnn 2019* (Berlin: Springer) pp 141–50
- Maaten L V D and Hinton G 2008 Visualizing data using t-SNE *J. Machine Learning Res.* **9** 2579–605
- McFarland D J, Parvaz M A, Sarnacki W A, Goldstein R Z and Wolpaw J R 2016 Prediction of subjective ratings of emotional pictures by EEG features *J. Neural Eng.* **14** 016009
- Meneses Alarcro S and Fonseca M J 2017 Emotions recognition using EEG signals: A survey *IEEE Trans. Affective Computing* **10** 374–93
- Mikolov T, Chen K, Corrado G and Dean J 2013 Efficient estimation of word representations in vector space arXiv:1301.3781
- Mirza M and Osindero S 2014 Conditional generative adversarial nets arXiv:1411.1784
- Mogren O 2016 C-RNN-GAN: Continuous recurrent neural networks with adversarial training arXiv:1611.09904
- Mühl C, Allison B, Nijholt A and Chanel G 2014 A survey of affective brain computer interfaces: principles, state-of-the-art and challenges *Brain-Computer Interfaces* **1** 66–84
- Perez L and Wang J 2017 The effectiveness of data augmentation in image classification using deep learning arXiv:1712.04621
- Petrantonakis P C and Hadjileontiadis L J 2010 Emotion recognition from EEG using higher order crossings *IEEE Trans. on Information Technology in Biomedicine* **14** 186–97
- Qi G J 2017 Loss-sensitive generative adversarial networks on lipschitz densities arXiv:1701.06264v6

- Radford A, Metz L and Chintala S 2016 Unsupervised representation learning with deep convolutional generative adversarial networks *Int. Conf. on Learning Representations* arXiv:1511.06434
- Salimans T, Kingma D and Welling M 2015 Markov chain monte carlo and variational inference: Bridging the gap *Int. Conf. on Machine Learning* pp 1218–26
- Samara A, Menezes M L R and Galway L 2017 Feature extraction for emotion recognition and modelling using neurophysiological data *Int. Conf. on Ubiquitous Computing Communications Int. Symp. on Cyberspace Security* pp 138–44
- Shi L C, Jiao Y Y and Lu B L 2013 Differential entropy feature for EEG-based vigilance estimation *Int. IEEE/ Conf. on Neural Engineering (NER)* pp 6627–30
- Simard P Y, Steinkraus D W and Platt J 2003 Best practices for convolutional neural networks applied to visual document analysis *Int. Conf. on Document Analysis and Recognition* pp 958–63
- Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv:1409.1556
- Smith D and Burke B 2019 Hype cycle for emerging technologies, 2019 (<https://www.gartner.com/en/documents/3956015/hype-cycle-for-emerging-technologies-2019>)
- Soleymani M, Lichtenauer J, Pun T and Pantic M 2012 A multimodal database for affect recognition and implicit tagging *IEEE Trans. Affective Computing* **3** 42–55
- Somers M 2019 Emotion AI, explained (<https://mitsloan.mit.edu/ideas-made-to-matter/emotion-ai-explained>)
- Song T, Zheng W, Lu C, Zong Y, Zhang X and Cui Z 2019 Mped: A multi-modal physiological emotion database for discrete emotion recognition *IEEE Access* **7** 12177–91
- Tanja B, Didier G and Scherer K R 2009 Emotion recognition from expressions in face, voice and body: the multimodal emotion recognition test (mert) *Emotion* **9** 691
- Wang F, Zhong S-H, Peng J-F, Jiang J-M and Liu Y 2018 Data augmentation for EEG-based emotion recognition with deep convolutional neural networks *Int. Conf. on Multimedia Modeling* pp 82–93
- Wang X-W, Nie D and Lu B-L 2014 Emotional state classification from eeg data using machine learning approach *Neurocomputing* **129** 94–106
- Wu J, Zhang C, Xue T, Freeman W T and Tenenbaum J B 2016 Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling *Neural Information Processing Systems* 82–90
- Yang Y, Wu Q M J, Zheng W-L and Lu B-L 2018 EEG-based emotion recognition using hierarchical network with subnetwork nodes *IEEE Trans. Cognitive Developmental Syst.* **10** 408–19
- Zhang Q, Yang L T, Chen Z and Peng L 2018 A survey on deep learning for big data *Information Fusion* **42** 146–57
- Zhao L-M, Li R, Zheng W-L and Lu B-L 2019 Classification of five emotions from EEG and eye movement signals: complementary representation properties *Int. IEEE/ Conf. on Neural Engineering (NER) IEEE* pp 611–14
- Zheng W-L, Liu W, Lu Y-F, Lu B-L and Cichocki A 2019 Emotionmeter: A multimodal framework for recognizing human emotions *IEEE Trans. Cybernetics* **49** 1110–22
- Zheng W-L and Lu B-L 2015 Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks *IEEE Trans. Autonomous Mental Development* **7** 162–75
- Zheng W-L, Zhu J-Y and Lu B-L 2019 Identifying stable patterns over time for emotion recognition from EEG *IEEE Trans. Affective Computing* **10** 417–29
- Zheng Z, Zheng L and Yang Y 2017 Unlabeled samples generated by gan improve the person re-identification baseline in vitro *Proc. of the IEEE Int. Conf. on Computer Vision* pp 3754–62
- Zhu X, Liu Y, Li J, Wan T and Qin Z 2018 Emotion classification with data augmentation using generative adversarial networks *Pacific- Conf. on Knowledge Discovery and Data Mining* pp 349–60
- van Dyk D A and Meng X-L 2001 The art of data augmentation *J. Computat. Graphical Stat.* **10** 1–50