Adversarial Variational Embedding for Robust Semi-supervised Learning

Xiang Zhang, Lina Yao, Feng Yuan
University of New South Wales, Sydney, Australia
xiang.zhang3@student.unsw.edu.au,lina.yao@unsw.edu.au,feng.yuan@student.unsw.edu.au

ABSTRACT

Semi-supervised learning is sought for leveraging the unlabelled data when labelled data is difficult or expensive to acquire. Deep generative models (e.g., Variational Autoencoder (VAE)) and semi-supervised Generative Adversarial Networks (GANs) have recently shown promising performance in semi-supervised classification for the excellent discriminative representing ability. However, the latent code learned by the traditional VAE is not exclusive (repeatable) for a specific input sample, which prevents it from excellent classification performance. In particular, the learned latent representation depends on a non-exclusive component which is stochastically sampled from the prior distribution. Moreover, the semi-supervised GAN models generate data from pre-defined distribution (e.g., Gaussian noises) which is independent of the input data distribution and may obstruct the convergence and is difficult to control the distribution of the generated data. To address the aforementioned issues, we propose a novel Adversarial Variational Embedding (AVAE) framework for robust and effective semi-supervised learning to leverage both the advantage of GAN as a high quality generative model and VAE as a posterior distribution learner. The proposed approach first produces an exclusive latent code by the model which we call VAE++, and meanwhile, provides a meaningful prior distribution for the generator of GAN. The proposed approach is evaluated over four different real-world applications and we show that our method outperforms the state-of-the-art models, which confirms that the combination of VAE++ and GAN can provide significant improvements in semi-supervised classification.

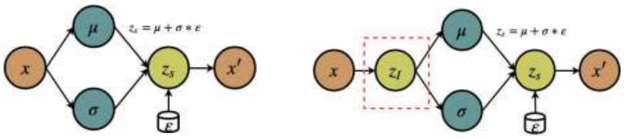
KEYWORDS

Variational Autoencoder, Generative Adversarial Networks, Representation Learning, Semi-supervised Classification

ACM Reference Format:
Xiang Zhang, Lina Yao, Feng Yuan. 2019. Adversarial Variational Embedding for Robust Semi-supervised Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330966

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330966



(a) Standard VAE

(b) VAE++

Figure 1: Comparison of the standard VAE and the proposed VAE++. x and x' denote the input and the reconstructed data. μ and σ denote the learned expectation and standard deviation, z_s denotes the stochastically sampled latent representation which is composed by μ, σ, and ε, where ε is randomly sampled from $\mathcal{N}(0, 1)$. In standard VAE, z_s is regarded as the learned representation while, in VAE++, z_I denotes the proposed exclusive latent representation which can be used for classification.

1 INTRODUCTION

Semi-supervised learning from data is one of the fundamental challenges in artificial intelligence, which considers the problem when only a subset of the observations has corresponding class labels [6]. This issue is of immense practical interest in a broad range of application scenarios, such as abnormal activity detection [32], neurological diagnosis [24], and computer vision [7]. In these scenarios, it is easy to obtain abundant observations but expensive to gather the corresponding class labels. Among existing approaches, Variational Autoencoders (VAEs) [12, 28] have recently achieved state-of-the-art performance in semi-supervised learning.

VAE models provide a general framework for learning latent representations: a model is specified by a joint probability distribution both over the data and over latent random variables, and a representation can be found by considering the posterior on latent variables given specific data [20]. The learned representations can not only be used for generation but also for classification. For instance, VAE provides a latent feature representation of the input observations, where a separate classifier can be thereafter trained using these representations. The high quality of latent representations enables accurate classification, even with a limited number of labels. A number of studies have applied VAE in semi-supervised classification in the computer vision area [12, 17, 20].

1.1 Motivation

Why we propose the VAE++ . One major challenge faced by the existing VAE-based semi-supervised methods is that the latent representations are stochastically sampled from the prior distribution instead of being directly rendered from the explicit observations. In particular, as shown in Figure 1a, the learned latent representations z_s are *randomly sampled* from a multivariate Gaussian distribution (see Equation 1). Thus, for a specific sample, the corresponding latent representation is not exclusive (i.e., the representation is not repeatable in different runnings), which makes it inappropriate for classification. To solve this problem, in the latent space, we propose a new variable z_I (see Figure 1b) which is directly learned from the input data. The exclusive latent code z_I is guaranteed to keep invariant for a specific input $\boldsymbol{x}$ in different runnings. The modified VAE is called VAE++. In addition, the learned expectation $\boldsymbol{\mu}$ only contains a part of information of the input observations, which is not enough to represent the observations in classification task, even though $\boldsymbol{\mu}$ is exclusive[1]. The comparison of performance among z_I, z_s and $\boldsymbol{\mu}$ will be presented in Section 4.

Why VAE++ needs the semi-supervised GAN. In the proposed VAE++, it is necessary to reduce the information loss between the two latent representations z_I and z_s to guarantee the learned z_I is representative. The commonly used constraints between two distributions (e.g., Kullback-Leibler divergence) can only utilize the information of the observations but fail to exploit the information of labels. In this paper, we use a novel approach to take advantage of both unlabelled and labelled data by jointly training the VAE++ and a semi-supervised GAN.

Why semi-supervised GAN needs the VAE++. GAN based approaches [22, 26] have recently shown promising results in semi-supervised learning. The semi-supervised GAN trains a generative model and a discriminator with inputs belonging to one of K classes. Different from the regular GAN, the semi-supervised GAN requires the discriminator to make a $K + 1$ class prediction with an extra class added, corresponding to the generated fake samples. In this way, the observations' properties can be used to improve decision boundaries and allow for more accurate classification than using the labelled data alone. However, the generated samples are sampled from pre-defined distribution (e.g., Gaussian noise) [2]. Such pre-defined prior distributions are often independent from the input data distributions and may obstruct the convergence and can not guarantee the distribution of the generated data. This drawback can be amended by gearing with VAE++ which can provide a meaningful prior distribution that can represent the distribution of the input data.

We introduce a recipe for semi-supervised learning, a robust Adversarial Variational Embedding (AVAE) framework, which learns the exclusive latent representations by combining VAE and semi-supervised GAN. To utilize the generative ability of GAN and the distribution approximating power of VAE, the proposed approach employs GAN to encourage VAE for the aim of learning the more robust and informative latent code. We present the framework in the context of VAE, adding a new exclusive code in latent space which is directly rendered from the data space. The generator in VAE++ also works as a generator of GAN. Both the exclusive code (marked as real) and the generated representation (marked as fake) are fed into the discriminator in order to force them to have similar distribution [18].

1.2 Contribution

Although a small set of models combining VAE and GAN have been previously explored, they are all focused on the generation perspective. To our knowledge, we are in the first batch of work that focuses on classification by aggregating VAE and GAN. We mark the following contributions:

- We present a novel semi-supervised Adversarial Variational Embedding approach to harness the deep generative model and generative adversarial networks collectively under a trainable unified framework. The reproducible codes and datasets are publicly available[2].
- We propose a new structure, VAE++, to automatically learn an exclusive latent code for accurate classification. A novel semi-supervised GAN, which exploits both the unlabelled data distribution and categorical information, is proposed to gear with the VAE++ in order to encourage the VAE++ to learn a more effective and robust exclusive code.
- We evaluate the proposed approach over four real-world applications (activity reconstruction, neurological diagnosis, image classification, and recommender system). The results demonstrate that our approach outperforms all the state-of-the-art methods.

2 RELATED WORK

There are a host of studies that have been investigated to apply VAE for semi-supervised learning [12, 16, 20, 28]. [12] explores semi-supervised learning with deep generative models by building two VAE-based deep generative models for latent representation extraction. Afterward, [20] attempts to learn disentangled representations that encode distinct aspects of the data into separate variables. However, in all the existing semi-supervised VAE models, the learned representations do not only depend on the posterior distribution but also on the latent random variables. It is necessary that learning the exclusive code which is only related to the posterior distribution for the specified data.

Another recent arising semi-supervised method is semi-supervised GAN [22, 29] . SGAN [22] extends GAN to the semi-supervised context by forcing the discriminator network to output class labels. The CatGAN [29] modifies the objective function to take into account the mutual information between observed examples and their predicted class distributions. In the above methods, the generator chooses simple factored continuous noise which is independent from the input data distribution, for generation. As a result, it is possible that the noise will be used by the generator in a highly entangled way, increasing the difficulty to control the distribution of the generated data. Conditional GAN [18] and InfoGAN address this drawback by utilizing external information (e.g., categorical information) as a restriction, but they both pay attention to generation or supervised classification and have limited help in semi-supervised classification.

Despite the few works attempting to combine VAE and GAN [17], most of them focus on generation instead of classification. For example, the VAE/GAN and CVAE-GAN employ the standard VAE to share the encoder with the generator of GAN in order to generate new observations. For semi-supervised classification, we care about the latent code instead of the observations. The Adversarial Autoencoder (AAE [17]) integrates VAE and GAN but

[1] For the same reason, $\boldsymbol{\sigma}$ can not be used as the exclusive code.

[2] https://github.com/xiangzhang1015/Adversarial-Variational-Semi-supervised-Learning

only employs GAN to replace KL divergence as a penalty to impose a prior distribution on the latent code, which is a totally different direction from our work.

Summary. Unlike the existing VAE- and GAN-based studies, the proposed model 1) focuses on semi-supervised classification instead of generation; 2) attempts to learn an exclusive latent representation instead of a stochastic sampled representation; 3) works on improvement of latent space instead of data space. Moreover, the semi-supervised GAN in our work partly adopts the improved GAN [26], but there are a number of differences: 1) [26] adopts the semi-supervised strategy for classification while we adopt this strategy as a constraint to reduce information loss in the transformation from z_I to z_s in order to force the proposed AVAE to learn a more robust and effective latent code; 2) [26] employs the discriminator of GAN as the classifier while we adopt an extra non-parametric classifier since the former has poor performance in our case (take the PAMAP2 dataset as an example, [26] and our model achieve the accuracy around 65% and 85%, respectively); 3) we employ weighted loss function to balance the significance of the unlabelled and labelled observations.

3 METHODOLOGY

Suppose the input dataset has two subsets, one of which contains labelled samples while the other contains unlabelled samples. In the former subset, the observations appear as pairs $(\boldsymbol{X}^L, \boldsymbol{Y}^L) = \{(\boldsymbol{x}_1^L, \boldsymbol{y}_1), (\boldsymbol{x}_2^L, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_{N_L}^L, \boldsymbol{y}_{N_L})\}$ with the i-th observation $\boldsymbol{x}_i^L \in \mathbb{R}^M$ and the corresponding one-hot label $\boldsymbol{y_i} \in \mathbb{R}^K$ where K denotes the number of classes. N_L denotes the number of labelled observations while M denotes the number of the observation dimensions. In the latter subset, only the observations $\boldsymbol{X}^U = \{\boldsymbol{x}_1^U, \boldsymbol{x}_2^U, \cdots, \boldsymbol{x}_{N_U}^U\}$ are available and N_U denotes the number of unlabelled observations $\boldsymbol{x}_i^U \in \mathbb{R}^M$. The total data size N equals to the sum of N_L and N_U. In terms of effective classification, we attempt to learn a latent representation with distinguishable information. Then the learned representations can be fed into a classifier for recognition. In this paper, we mainly focus on the latent code learning.

In the semi-supervised learning, due to the lack of labelled observations, it is significant to learn latent variable distribution based on the observations without label[3]. Thus, we are required to build an encoder to provide an embedding or feature representation which allows accurate classification even with limited observations.

3.1 VAE++

The VAE is demonstrated to provide a latent feature representation for semi-supervised learning [12, 20], compared to a linear embedding method or a regular autoencoder. The VAE maps the input observation $\boldsymbol{x}$ to a compressed code z_s, and decodes it to reconstruct the observation. The latent representation is calculated through the reparameterization trick [13]:

$$z_s = \mu_{\boldsymbol{x}} + \sigma_{\boldsymbol{x}} * \boldsymbol{\varepsilon} \tag{1}$$

with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1)$ to impose the posterior distribution of the latent code on $p(z_s|x) \sim \mathcal{N}(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}^2)$. $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$ denote the expectation and standard deviation of the posterior distribution of z_s, which are learned from $\boldsymbol{x}$. For the efficient generation and reconstruction, VAE imposes the code z_s on a prior Gaussian distribution:

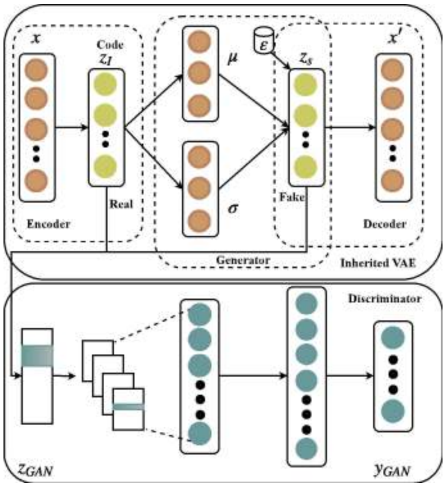


Figure 2: AVAE is composed of VAE++ and a semi-supervised GAN. The generated z_s (labelled as fake) and the exclusive code z_I (labelled as real) are fed into the discriminator. The discriminator can exploit both the labelled and unlabelled observations. The generator in VAE++ also works as a generator of GAN.

$$\bar{p}(z_s) = \mathcal{N}(z_s|0, \boldsymbol{I})$$

Through minimizing the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{x'}$ and restricting the distribution of z_s to approximate the prior distribution $\bar{p}(z_s)$, VAE is supposed to learn the representative latent code z_s which can be used for classification or generation.

Due to the strong feature representation ability, VAE has been employed for feature extraction and semi-supervised learning [20, 30]. However, one limitation of the standard VAE is that the learned latent code $z_s = g(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}, \boldsymbol{\varepsilon})$, as shown in Equation (1), is not exclusive. In other words, for a specific observation $\boldsymbol{x}$ and a fixed embedding model $p(z_s|\boldsymbol{x})$, the corresponding latent code z_s is not exclusive as it contains a stochastic variable $\boldsymbol{\varepsilon}$ which is randomly sampled from the prior distribution $\bar{p}(z_s)$. For instance, in a pretrained fixed VAE encoder, the specific input $\boldsymbol{x}$ will lead to a variety of z_s in different running. At high level, the latent code z_s is determined by two factors: the prior distribution of observation $\bar{p}(\boldsymbol{x})$ which affects z_s through the learned $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$, and the stochastically sampled data $\boldsymbol{\varepsilon}$. However, the stochastically sampled latent code is unstable and will corrupt the features for classification. Furthermore, the posterior distribution of z_s is forced to approximate the manually set prior distribution (commonly Normal Gaussian distribution), which inevitably leads to information loss.

In order to completely sidestep the above-mentioned issue, in this paper, we propose a novel VAE++ model to learn an exclusive latent code z_I. The VAE++ contains three key components: the encoder, the generator, and the decoder (see Figure 2). The encoder transforms the observation into a latent code $z_I \in \mathbb{R}^D$ which is directly determined by the input $\boldsymbol{x}$. D denotes the dimension of z_I. We learn the:

$$p_{\theta_{en}}(z_I|\boldsymbol{x}) = f(z_I; \boldsymbol{x}, \boldsymbol{\theta}_{en})$$

[3] For simplification, we omit the index and directly use variable $\boldsymbol{x}$ to denote observations.

where f denotes a non-linear transformation while $\boldsymbol{\theta}_{en}$ denotes encoder parameters. The non-linear transformation f is generally chosen as a deep neural network for the excellent ability of non-linear approximation. Then, in the generator, we measure the expectation $\mu(z_I)$ and the standard derivation $\sigma(z_I)$ from the latent code z_I and update Equation (1). The generated variable z_s can be calculated by:

$$z_s = \mu(z_I) + \sigma(z_I) * \boldsymbol{\varepsilon} \tag{2}$$

At last, the decoder is employed to reconstruct the sample:

$$p_{\boldsymbol{\theta}_{de}}(\boldsymbol{x}'|z_s) = f'(\boldsymbol{x}'; z_s, \boldsymbol{\theta}_{de})$$

where f' denotes another non-linear rendering, called decoder, with parameters $\boldsymbol{\theta}_{de}$ and $\boldsymbol{x}'$ denotes the reconstructed observation.

The loss function of VAE++ can be calculated by:

$$\begin{aligned}\mathcal{L}_{VAE} = &-\mathbb{E}_{z_s \sim p_{\boldsymbol{\theta}_{en}}(z_s|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}_{de}}(\boldsymbol{x}'|z_s)] \\ &+ KL(p_{\boldsymbol{\theta}_{en}}(z_s|\boldsymbol{x})||\bar{p}(z_s))\end{aligned} \tag{3}$$

The first component is the reconstruction loss, which equals to the expected negative log-likelihood of the observation. This term encourages the decoder to reconstruct the observation $\boldsymbol{x}$ based on the sampling code z_s which is under Gaussian distribution. The lower reconstruction error indicates the encoder learned a better latent representation. The second component is the Kullback-Leibler divergence which measures the distance between the prior distribution of the latent code $\bar{p}(z_s)$ and the posterior distribution $p(z_s|\boldsymbol{x})$. This divergence reflects the information loss when we use $p(z_s|\boldsymbol{x})$ to represent $\bar{p}(z_s)$.

In the latent space of the novel VAE++, there are two compressed informative codes z_I and z_s. The former represents directly-encoded $\boldsymbol{x}$ whilst the latter is stochastically sampled from the posterior distribution , which makes the former more suitable for classification. Therefore, we choose z_I as the compressed latent code in VAE++ instead of the z_s in standard VAE.

From equation (2), we can observe that the expectation and standard deviation of z_s and z_I are invariant. In particular, for a specific sample $\boldsymbol{x}_i$, the corresponding z_{si} and z_{Ii} have the same statistical characteristics. Thus, we have

$$z_s \longleftarrow \mu(z_I), \sigma(z_I), \boldsymbol{\varepsilon}$$

which indicates that the generated z_s is affected by both the distribution of z_I and the prior distribution $\bar{p}(z_s)$ (or $\boldsymbol{\varepsilon}$). In summary, the z_s inherits the statistical characteristics of z_I.

3.2 Adversarial Variational Embedding

One significant sufficient condition of a well-trained VAE++ is less information loss in the transformation from z_I to z_s to guarantee the learned z_I is representative. As mentioned before, the information in z_s is partly inherited from z_I and the other part is randomly sampled from the prior distribution $\bar{p}(z_s)$. Since the conditional distribution $p_{\boldsymbol{\theta}_{en}}(z_I|\boldsymbol{x})$ has a better description of the input observation $\boldsymbol{x}$, we attempt to increase the proportion of inherited part and decrease the proportion of stochastically sampled part.

As shown in Figure 2, in the proposed AVAE the generator $\boldsymbol{G}$ generates z_s based on the joint probability $p(\mu, \sigma, \bar{p}(z_s))$ instead of the noise in standard GAN. The z_s is regarded as 'fake' while z_I is marked as 'real'. Specifically, for the labelled observations $\boldsymbol{x}^L$, VAE++ encodes the input to the latent code $z_I^L \in \mathbb{R}^D$ and generates $z_s^L \in \mathbb{R}^D$; similarly, for unlabelled observations $\boldsymbol{x}^U$, we have $z_I^U \in \mathbb{R}^D$ and generates $z_s^U \in \mathbb{R}^D$. To exploit the information of the labels, we extend the $\boldsymbol{y} \in \mathbb{R}^K$ which has K possible classes to $\boldsymbol{y}_{GAN} \in \mathbb{R}^{K+1}$ which has $K+1$ possible classes by regarding the generated fake samples z_s as the $(K+1)$-th class [22, 26]. In the VAE++, the unspecified z_s denotes both z_s^L and z_s^U whenever we don't care whether the observation is labelled or not. This rule also applies to z_I. Similarly, we use z_{GAN} to denote the input of the discriminator $\boldsymbol{D}$, which contains both z_I and z_s. The discriminator can be described by

$$\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN}|z_{GAN}) = h(\boldsymbol{y}_{GAN}; z_{GAN}, \boldsymbol{\varphi})$$

where $\boldsymbol{\varphi}$ denotes the parameters of $\boldsymbol{D}$ while h denotes the non-linear transformation which is implemented by a Convolutional Neural Networks (CNN) [14] in this paper. Therefore, we can use $\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|z_{GAN})$ to supply the probability where z_{GAN} is fake (from z_s) and use $\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN}|z_{GAN}, \boldsymbol{y}_{GAN} < K+1)$ to supply the probability where z_{GAN} is real ((from z_I)) and is correctly classified.

For the labelled input, same as supervised learning, the discriminator is supposed to not only tell whether the input z_{GAN} is real or generated, but also classify it into the correct class. Therefore, we have the supervised loss function

$$\mathcal{L}_{label} = -\mathbb{E}_{z_{GAN}, \boldsymbol{y}_{GAN} \sim p_j}[log \boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN}|z_{GAN}, \boldsymbol{y}_{GAN} < K+1)]$$

where p_j denotes the joint probability.

For the unlabelled input, we only require the discriminator to perform a binary classification: the input is real or fake. The former probability can be calculated by $1 - \boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|z_{GAN})$ whilst the latter can be calculated by $\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|z_{GAN})$. Thus, the unsupervised loss function:

$$\begin{aligned}\mathcal{L}_{unlabel} = &-\mathbb{E}_{z_{GAN} \sim p_{\boldsymbol{\theta}_{en}}(z_I|\boldsymbol{x})}[log(1 - \boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|z_{GAN}))] \\ &- \mathbb{E}_{z_{GAN} \sim p_{\boldsymbol{\theta}_{en}}(z_s|\boldsymbol{x})}[log(\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|z_{GAN}))]\end{aligned}$$

In summary, the final loss function of the discriminator

$$\mathcal{L}_{GAN} = w_1 * flag * \mathcal{L}_{label} + w_2 * (1 - flag) * \mathcal{L}_{unlabel} \tag{4}$$

where w_1, w_2 are weights and $flag$ is a switch function

$$flag = \begin{cases} 1 & labelled \\ 0 & unlabelled \end{cases}$$

If the specific observation is labelled, we calculate the labelled loss function. Otherwise, we calculate the unlabelled loss function. From empirical experiments, we observe that the $\mathcal{L}_{unlabel}$ is much easier to converge than $\mathcal{L}_{label}$ and the real/fake classification accuracy is much higher than the K classes classification accuracy. To encourage the optimizer to focus on the former part which is more difficult to converge, we set $w_1 = 0.9$ and $w_2 = 0.1$.

The discriminator receives z_{GAN} as input and extracts the dependencies through CNN filters. Two fully connected layers follow the convolutional layer for dimension reduction. At last, a softmax layer is employed to work on the low-dimension features to estimate the log normalization of the categorical probability distribution which is output as $\boldsymbol{y}_{GAN}$.

The overall aim of the proposed AVAE (as described in Algorithm 1) is to train a robust and effective semi-supervised embedding method. The VAE loss $\mathcal{L}_{VAE}$ and the GAN loss $\mathcal{L}_{GAN}$ are trained

Algorithm 1: Adversarial Variational Embedding

```
Input: labelled observations ($\boldsymbol{X}^L$, $\boldsymbol{Y}^L$) and unlabelled observations $\boldsymbol{X}^U$
Output: Representation $\boldsymbol{z_I}$
Initialize network parameters $\boldsymbol{\theta_{en}}$, $\boldsymbol{\theta_{de}}$, $\boldsymbol{q_\varphi}$
for $\boldsymbol{x} \in \{X^L, X^U\}$ do
    $\boldsymbol{z_I} \leftarrow \boldsymbol{x}$
    $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \boldsymbol{z_I}$
    Sampling $\boldsymbol{\varepsilon}$ from $\mathcal{N}(0, \boldsymbol{I})$
    $\boldsymbol{z_s} = \mu(z_I) + \sigma(z_I) * \varepsilon$
    $\boldsymbol{x'} \leftarrow \boldsymbol{z_s}$
    $\mathcal{L}_{VAE} \leftarrow \boldsymbol{x}, \boldsymbol{x'}, p(\boldsymbol{z_s}|\boldsymbol{x})$
    for $z_I, z_s, \boldsymbol{y} \in Y^L$ do
        $\boldsymbol{y}_{GAN} \leftarrow z_I, z_s$
        $\mathcal{L}_{GAN} \leftarrow \boldsymbol{y}_{GAN}, \boldsymbol{y}$
    end for
    Minimize $\mathcal{L}_{VAE}$ and $\mathcal{L}_{GAN}$
end for
return $\boldsymbol{z_I}$
```

simultaneously by the Adam optimizer. After convergence, the compressed representative code z_I is fed into a non-parametric nearest neighbors classifier for recognition.

4 EXPERIMENTS

In this section, we demonstrate the effectiveness and validation of the proposed method over four applications.

4.1 Activity Recognition

4.1.1 Experiment Setup. Activity recognition is an important area in data mining. We evaluate our approach over the well-known PAMAP2 dataset [5], which is collected by 9 participants (8 males and 1 female) aged 27 ± 3. We select 5 most commonly used activities (Cycling, standing, walking, lying, and running, labelled from 0 to 4) as a subset for evaluation. For each subject, there are 12,000 instances. The activity is measured by 3 Inertial Measurement Units (IMU) attached to the participants' wrist, chest, and the outer ankle. Each IMU includes 13 dimensions: two 3-axis accelerometers, one 3-axis gyroscopes, one 3-axis magnetometers and one thermometer. The experiments are performed by a Leave-One-Subject-Out strategy to ensure the practicality.

The time window is set as 10 with 50% overlapping. The dataset is split into a training set (80% proportion) and a testing set (20% proportion). For semi-supervised learning, the training dataset contains both labelled observations and unlabelled observations. We present a term called'supervision rate' as a handle on the relative weight between the supervised and unsupervised terms. For the given number of labelled observations N^L and the number of unlabelled observations N^U, the supervision rate γ is defined by $N^L/(N^L + N^U)$.

4.1.2 Parameter Setting. We introduce the default parameter settings and the settings in other applications keep the same if not mentioned. The input observations are first normalized by Z-score normalization and fed to the input layer of the unsupervised VAE++. The neuron amount in the first hidden layer, which is denoted by z_I, is a quarter of M. The second hidden layer contains 2 components which represent the expectation and the standard deviation respectively. The third hidden layer z_s has the sample shape with z_I. An Adam optimizer with a learning rate of 0.00001 is employed to minimize the loss function of VAE++.

After each epoch of VAE++, the first hidden layer z_I and the third hidden layer z_s are labelled as 'real' and 'fake', respectively, and fed to the discriminator $\mathcal{D}$. The discriminator contains one convolutional layer followed by two fully-connected layers. There is a softmax layer to obtain the categorical probability before the output layer which has $K+1$ neurons. The convolutional layer has 10 filters which have shape [2, 2] and the stride size [1, 1]. The padding method of the convolutional operation is set as 'same' while the activation function is ReLU. The following hidden layer has $M/4$ neurons and the sigmoid activation function. The loss function is optimized by Adam update rule with learning rate of 0.0001. The object functions of the VAE++ and the discriminator are trained simultaneously. After the convergence, the semi-supervised learned latent representation z_I is fed into a supervised non-parametric nearest neighbor classifiers with $k = 3$.

4.1.3 Baselines. To measure the effectiveness of the proposed method, we compare it with a set of competitive state-of-the-art models. The state-of-the-art methods are composed of two categories: algorithm-related and application-related. The former denotes other VAE/GAN based semi-supervised classification algorithms, which are the same for all the applications. The comparison is used to demonstrate our framework has the highest semi-supervised representation learning ability. The latter denotes the state-of-the-art models in each application, which are varied for the different applications. The comparison is used to demonstrate our work is effective in the real-world scenarios.

The algorithm-related semi-supervised learning solutions in our comparison are listed as follows:

- M2. [12] proposes a probabilistic model that describes the data as being generated by a latent class variable in addition to a continuous latent representation.
- Adversarial Autoencoders (AAE). [17] employs the GAN to perform variational inference by matching the aggregated posterior of the hidden representation of the autoencoder.
- Ladder Variational Autoencoders (LVAE). [28] proposes an inference model which recursively corrects the generative distribution by a data dependent likelihood.
- Auxiliary Deep Generative Models (ADGM). [16] extends deep generative models with auxiliary variables, which improves the variational approximation.

We design ablation study to demonstrate the necessity of each key component of the proposed approach. In the ablation study, we set four control experiments with single variable among the components of AVAE. We adopt the following four methods to discover the latent representations: 1) VAE ($\boldsymbol{\mu}$) with $\boldsymbol{\mu}$ as the latent representation; 2) standard VAE (z_s as the latent representation); 3) VAE++ (z_I as the latent representation); 4) AVAE. The extracted representations are fed into the same classifier for classification.

The application-related state-of-the-art models on activity recognition are listed here:

Table 1: Overall comparison of semi-supervised classification accuracy (%) on activity recognition. All the baselines and our approach are working on the same dataset and sharing the same experiment settings for each specific application.

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[4]	[15]	[9]	[33]	VAE (μ)	VAE	VAE++	AVAE
Activity Recognition (PAMAP2)	20	64.83±0.16	63.67±0.23	69.82±0.69	67.31±0.45	72.31±0.16	70.95±0.08	67.31±0.14	76.68±0.31	58.43±0.13	76.51±0.53	78.12±0.55	78.63±0.38
	40	68.92±0.23	76.83±0.25	76.43±0.19	78.21±0.38	80.51±0.21	75.38±0.12	77.28±0.21	80.15±0.16	62.74±0.12	78.78±0.22	80.88±0.38	81.37±0.29
	60	72.35±0.21	77.39±0.19	78.69±0.27	79.34±0.29	80.29±0.21	76.89±0.05	79.69±0.15	82.49±0.33	67.85±0.08	79.63±0.29	81.94±0.19	84.91±0.17
	80	75.88±0.35	78.28±0.11	81.41±0.23	80.38±0.16	82.12±0.16	79.95±0.18	81.65±0.09	83.56±0.11	73.43±0.06	81.75±0.17	82.08±0.26	85.56±0.21
	100	77.59±0.17	80.79±0.14	84.39±0.18	83.66±0.16	83.64±0.12	81.96±0.11	82.38±0.13	84.59±0.24	76.85±0.00	82.37±0.25	83.29±0.18	86.41±0.06

Note: If the compared method can not deal with unsupervised samples, it will be trained only by the supervised samples.

Table 2: Overall comparison of semi-supervised classification accuracy (%) on neurological diagnosis

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[34]	[10]	[27]	[8]	VAE (μ)	VAE	VAE++	AVAE
Neurological Diagnosis (TUH)	20	71.28±0.16	80.13±0.95	82.31±0.19	86.32±0.12	87.66±0.23	86.38±0.36	82.19±0.24	86.33±0.21	80.58±0.69	86.37±0.24	0.86±0.53	93.69±0.16
	40	75.32±0.16	82.95±0.26	84.38±0.16	86.99±0.05	89.25±0.19	91.58±0.35	84.21±0.08	89.25±0.34	81.35±0.24	89.69±0.27	91.28±0.25	94.32±0.28
	60	76.32±0.29	86.21±0.52	87.51±0.26	87.65±0.16	91.28±0.37	92.58±0.26	85.36±0.32	90.38±0.24	82.59±0.63	90.58±0.27	92.87±0.31	95.21±0.21
	80	79.65±0.37	88.53±0.28	89.56±0.25	88.05±0.12	92.59±0.26	93.25±0.31	85.16±0.24	91.59±0.16	83.21±0.21	91.69±0.35	93.96±0.28	97.86±0.26
	100	82.59±0.31	89.58±0.25	90.25±0.21	88.65±0.26	93.32±0.18	94.29±0.25	86.42±0.26	92.4±0.25	84.21±0.65	92.38±0.41	94.65±0.24	98.13±0.32

Table 3: Overall comparison of semi-supervised classification accuracy (%) on image classification

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[22]	[29]	[31]	[19]	VAE (μ)	VAE	VAE++	AVAE
Image Classification (MNIST)	20	93.22±0.62	90.25±0.25	93.25±0.26	89.61±0.27	95.23±0.34	94.25±0.13	94.58±0.25	92.96±0.28	91.58±0.24	92.31±0.53	93.59±0.31	95.12±0.19
	40	93.25±0.34	93.21±0.23	93.28±0.46	91.58±0.25	95.27±0.53	95.56±0.08	95.21±0.26	93.21±0.56	93.65±0.21	94.21±0.19	94.68±0.28	96.43±0.35
	60	96.24±0.51	96.35±0.27	95.34±0.21	93.21±0.34	96.38±0.22	96.54±0.08	96.48±0.32	96.28±0.57	94.89±0.21	95.34±0.14	96.42±0.25	97.21±0.21
	80	98.19±0.25	95.32±0.37	96.11±0.52	95.01±0.15	97.82±0.11	97.21±0.13	97.86±0.34	97.63±0.15	96.78±0.25	97.63±0.15	98.71±0.16	99.79±0.12
	100	98.65±0.21	0.98.25±0.61	96.35±0.26	95.38±0.82	99.21±0.26	98.64±0.27	99.06±0.22	98.53±0.17	97.41±0.18	98.35±0.09	99.67±0.23	99.85±0.11

Table 4: Overall comparison of semi-supervised classification accuracy (%) on recommender system

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[23]	[25]	[11]	[3]	VAE (μ)	VAE	VAE++	AVAE
Recommender System (Yelp)		66.42±0.17	58.27±0.35	66.35±0.36	54.27±0.38	40.55±0.27	47.58±0.36	65.99±0.62	66.21±0.24	64.28±0.12	64.39±0.62	65.58±0.37	70.19±0.87
	20	69.36±0.37	61.55±0.62	68.16±0.24	55.35±0.26	40.28±0.32	48.65±0.27	67.53±0.31	66.59±0.29	64.37±0.25	67.23±0.95	71.05±0.29	72.21±0.35
	40	72.58±0.19	62.15±0.39	68.59±0.93	57.63±0.23	42.15±0.16	50.95±0.24	66.58±0.29	67.95±0.38	67.56±0.35	69.58±0.37	72.19±0.62	75.34±0.35
	60	72.39±0.64	62.89±0.62	74.28±0.37	58.34±0.15	43.21±0.15	52.15±0.38	67.65±0.31	68.23±0.15	69.25±0.18	71.39±0.56	73.21±0.58	78.54±0.38
	80	74.58±0.62	63.51±0.86	72.59±0.36	59.58±0.23	45.86±0.22	54.10±0.12	68.03±0.17	70.61±0.25	73.24±0.68	73.28±0.69	76.53±0.28	79.38±0.59

- Chen et al. [4] adopt an attention mechanism to select the most distinguishable features from the activity signals and send them to a CNN structure for classification.
- Lara et al. [15] apply both statistical and structural detectors features to discriminate among activities.
- Guo et al. [9] exploit the diversity of base classifiers to construct a good ensemble for multimodal activity recognition, and the diversity measure is obtained from both labelled and unlabelled data.
- Zhang et al. [33] combine deep learning and the reinforcement learning scheme to focus on the crucial dimensions of the signals.

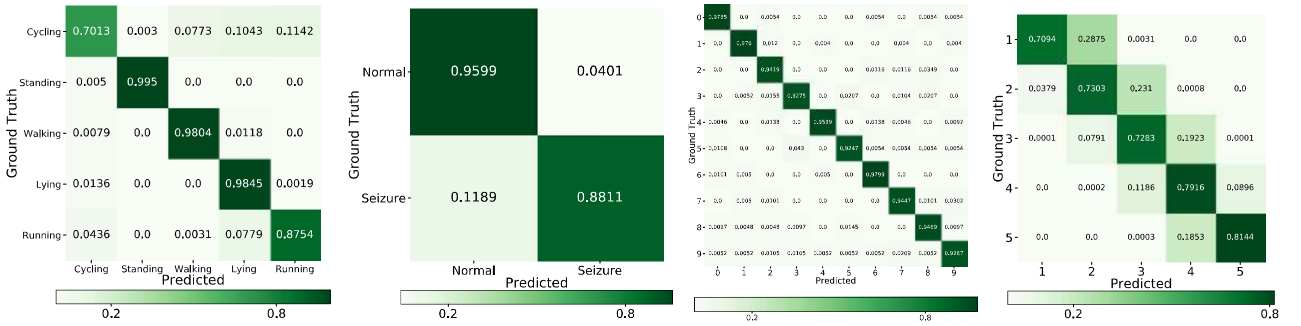
4.1.4 Results and Discussion. First, we report the overall performance of all the compared algorithms. From Table 1, we can observe that the proposed approach (AVAE) outperforms all the algorithm-related and application-related state-of-the-art models, illustrating the effectiveness of the latent space in providing robust representations for easier semi-supervised classification. The advantage is demonstrated under all the supervision rates.

In Table 1, through the ablation study, it is observed that each component (especially GAN) contributes to the performance enhancement. Additionally, the proposed AVAE achieves a significant improvement which yields around 5% and 3% growth than the standard VAE and the VAE++ (under 60% supervision rate), respectively. This observation demonstrates that the proposed latent layer z_I and the adversarial training (between the discriminator and VAE++) encourages the proposed model to learn and refine the informative latent code. Take 60% supervision rate as an example, more details of the classification are shown in the confusion matrix (Figure 3a) and ROC curves with AUC score (Figure 4a).

4.2 Neurological Diagnosis

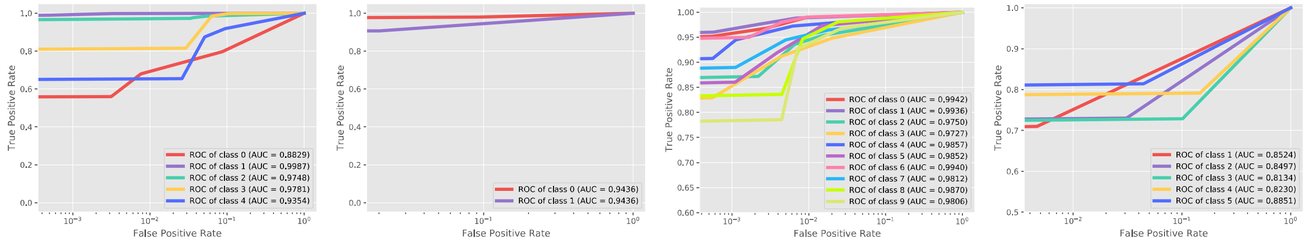
4.2.1 Experiment Setup. EEG signal collected in the unhealthy state differs significantly from the ones collected in the normal state [1]. The epileptic seizure is a common brain disorder that affects about 1% of the population and its octal state could be detected by the EEG analysis of the patient. In this application, we evaluate our framework with raw EEG data to diagnose the epileptic seizure of the patient.

We choose the benchmark dataset TUH [21] for epileptic seizure diagnosis. The TUH is a neurological seizure dataset of clinical EEG recordings associated with 22 channels from a 10/20 configuration. The sampling rate is set as 250 Hz. We select 12,000 samples from each of 18 subjects. Half of the samples are labelled as epileptic



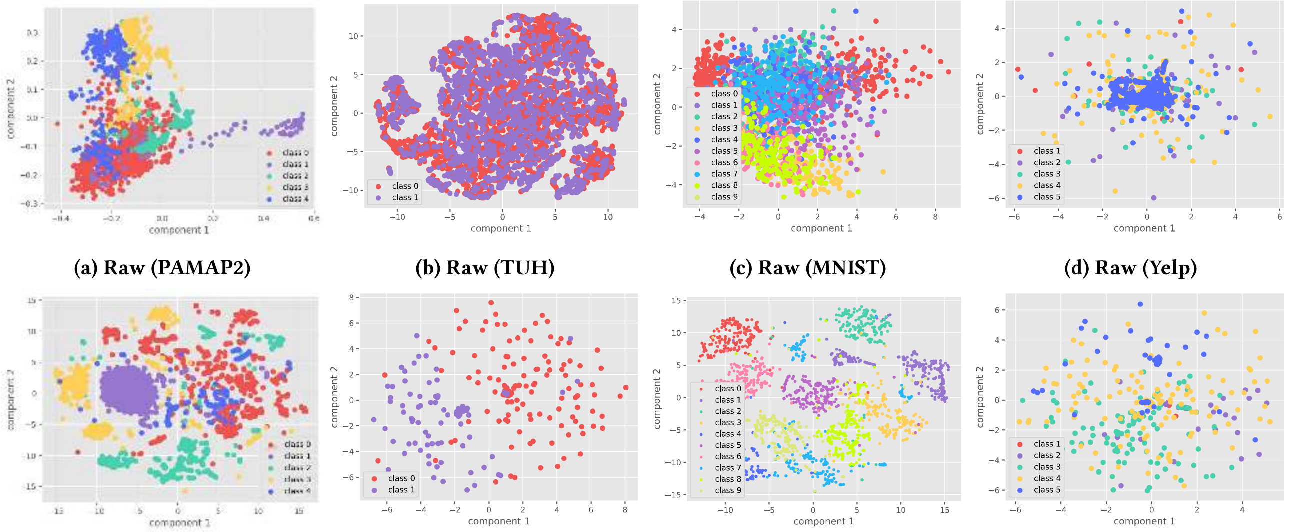
(a) Confusion matrix of PAMAP2 **(b) Confusion matrix of TUH** **(c) Confusion matrix of MNIST** **(d) Confusion matrix of Yelp**

Figure 3: Confusion matrix of PAMAP2, TUH, MNIST, and Yelp datasets.



(a) PAMAP2 **(b) TUH** **(c) MNIST** **(d) Yelp**

Figure 4: ROC curves of PAMAP2, TUH, MNIST, and Yelp datasets. The X-axis is in logarithmic scale.



(a) Raw (PAMAP2) **(b) Raw (TUH)** **(c) Raw (MNIST)** **(d) Raw (Yelp)**

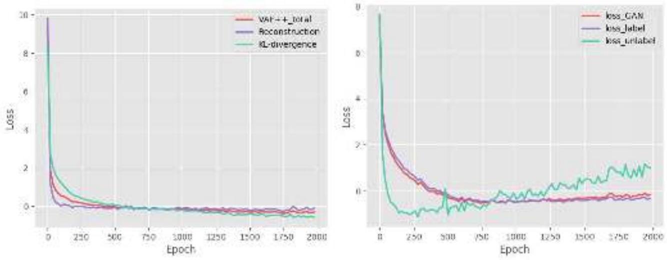
(e) Learned feature (PAMAP2) **(f) Learned feature (TUH)** **(g) Learned feature (MNIST)** **(h) Learned feature (Yelp)**

Figure 5: Visualization comparison between raw data and the semi-supervised learned features

seizure state (labelled as 1) and the remaining samples are labelled as normal state (labelled as 0). The experiment and parameter settings are the same as the activity recognition applications.

4.2.2 Baselines. The application-related state-of-the-art approaches in neurological diagnosis are listed here:

- Ziyabari et al. [34] adopt a hybrid deep learning architecture, including LSTM and stacked denoising Autoencoder, which integrates temporal and spatial context to detect the seizure.
- Harati et al. [10] demonstrate that a variant of the filter bank-based approach, coupled with first and second derivatives, provides a reduction in the overall error rate.



(a) VAE++

(b) GAN

Figure 6: Convergence curve of the VAE++ and GAN

- Schimeister et al. [27] attempt to improve the performance of seizure detection by combining deep ConvNets with training strategies such as exponential linear units.
- Goodwin et al. [8] combine RNN with access to textual data in EEG reports in order to automatically extracting word- and report-level features and infer underspecified information from EHRs (electronic health records).

4.2.3 Results and Discussion. From Table 2, we can observe that our approach outperforms all the competitive baselines on TUH dataset. For instance, under 60% supervision level, the proposed approach achieves the highest accuracy of 95.21% which claims around 4% improvement over other methods. The corresponding confusion matrix (Figure 3b) and ROC curves (Figure 4b) infer that the normal state has higher accuracy than the seizure state. One possible reason is that the start and end stage of the seizure has similar symptoms with the normal state which may lead to misclassification.

4.3 Image Classification

4.3.1 Experiment Setup. To evaluate the representation learning ability in images, we test our approach on the benchmark dataset MNIST [4]. MNIST contains 60,000 handwritten digital images (50,000 for training and 10,000 for testing) with $28 * 28$ pixels. The labels of this dataset are from 0 to 9, corresponding to the 10 digits.

4.3.2 Parameter Settings. Images are more informative compared to other application scenarios. The encoder of AVAE is designed to be stacked by two convolutional layers. The first convolutional layer has 32 filters with shape $[3, 3]$, the stride size $[1, 1]$, 'SAME' padding, and ReLU activation function. The followed pooling layer has $[2, 2]$ window size, $[2, 2]$ stride, and 'SAME' padding. The second convolutional layer has 64 filters with $[5, 5]$. The residual parameters of the second convolutional layer and pooling layer are the same with the former. Similarly, the decoder contains two de-convolutional layers with the same parameter settings.

4.3.3 Baselines. We reproduce the following methods under different supervision rate for comparison:

- Augustus [22] proposes a semi-supervised GAN (SGAN) by forcing the discriminator network to output class labels.
- Springenberg [29] proposes CatGAN to modify the objective function taking into account the mutual information between observation and the prediction distribution.
- Weston et al. [31] apply kernel methods for a nonlinear semi-supervised embedding algorithm.
- Miyato et al. [19] propose a regularization method based on virtual adversarial loss: a new measure of local smoothness of the conditional label distribution given the inputs.

4.3.4 Results and Discussion. As shown in Table 3, AVAE outperforms the counterparts with a slight gain with the same supervision level. The confusion matrix and ROC curves are reported in Figure 3c and Figure 4c. The results show that our approach is enabled to automatically learn the discriminative features by joint training the VAE++ and the semi-supervised GAN.

4.4 Recommender System

4.4.1 Experiment Setup. We apply our framework on recommender system scenarios, in particular, a restaurant rating prediction task based on the widely used Yelp dataset.

The Yelp Dataset[5] includes 192,609 Businesses, 1,637,138 Users, and 6,685,900 Ratings. Each business has 13 attributes (like 'near garage?', 'have valet?') which can describe the quality and convenience of the business. Meanwhile, each business is rated by a series customers. The ratings range from 1 to 5, which can reflect the customers' satisfactory degree. Our recommender task considers a unseen business's attributes as input data and predict the possible ratings from the potential customers. If the rating is high enough, the new business will be recommended to the public.

4.4.2 Baselines. We compare our approach with the state-of-the-art recommender system models which exploit the content information of items. Since these methods are used to make rating predictions for each user-item pair, we select those users who have 200 and more ratings in the Yelp dataset, generating a set of 1,111 users. After collecting the predicted ratings for all user-item pairs, we take the average item ratings over the users, which are further rounded to serve as the predicted labels.

- Pazzani et al. [23] summarizes basic content-based recommendation approaches, from which we select the cosine similarity-based nearest neighbour method as our fundamental baseline.
- Rendle [25] proposes the original implementation of factorization machine(FM) which is capable of incorporating item features with explicit feedbacks. We concatenate only the item indication vector and its feature after each user indication vector following the format in [25].
- He et al. [11] enhances the original FM using deep neural networks to learn high-order interactions between different item features.
- Chen et al. [3] applies feature- and item-level attention on item features, which is capable of emphasizing on the most important features.

4.4.3 Results and Discussion. From Table 4, we can observe that our approach outperforms both the competitive semi-supervised algorithms and the content-based recommender system state-of-the-art methods. The rating prediction details can be found in Figure 3d and Figure 4d. The classification performance is not good as in

[4] http://yann.lecun.com/exdb/mnist/

[5] https://www.yelp.com/dataset

other applications. One possible reason is that the attributes data are very sparse. The experiment results illustrate that our approach is effective in recommender system scenarios.

4.5 Further Analysis

Supervision Rate. We conduct extensive experiments to investigate the impact of supervision rate λ. The supervision rate ranges from 20% to 100% with 20% interval and each setting runs for at least three times with the average accuracy recorded. From Table 2 to Table 4, it is noticed that the proposed model obtains competitive performance at each supervision level.

Visualization. Figure 5 visualizes the raw data and the learned features on different datasets. The visualization comparison demonstrates the capability of our approach for feature learning.

Convergence. Take PAMAP2 as an example, Figure 6 presents the relationship between the loss function values and epoch numbers. The VAE++ loss includes the reconstruction loss and the KL-divergence whilst the loss of the discriminator in GAN includes labelled loss and unlabelled loss (with weights 0.9 and 0.1, respectively). We can observe that the proposed method shows good convergence property as it stablizes in around 200 epochs.

5 CONCLUSION

In this paper, we present an effective and robust semi-supervised latent representation framework, AVAE, by proposing a modified VAE model and integration with generative adversarial networks. The VAE++ and GAN share the same generator. In order to automatically learn the exclusive latent code, in the VAE++, we explore the latent code's posterior distribution and then stochastically generate a latent representation based on the posterior distribution. The discrepancy between the learned exclusive latent code and the generated latent representation is constrained by semi-supervised GAN. The latent code of AVAE is finally served as the learned feature for classification. The proposed approach is evaluated on four real-world applications and the results demonstrate the effectiveness and robustness of our model.

6 ACKNOWLEDGEMENT

This research was partially supported by grant ONRG NICOP N62909-19-1-2009.

REFERENCES

[1] Hojjat Adeli, Samanwoy Ghosh-Dastidar, and Nahid Dadmehr. A wavelet-chaos methodology for analysis of EEGs and EEG subbands to detect seizure and epilepsy. *IEEE Transactions on Biomedical Engineering* 54, 2 (????).

[2] Jiezhang Cao, Yong Guo, Qingyao Wu, Chunhua Shen, and Mingkui Tan. 2018. Adversarial Learning with Local Coordinate Coding. *The International Conference of Machine Learning (ICML)* (2018).

[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. ACM.

[4] Kaixuan Chen, Lina Yao, Xianzhi Wang, Dalin Zhang, Tao Gu, Zhiwen Yu, and Zheng Yang. 2018. Interpretable Parallel Recurrent Neural Networks with Convolutional Attentions for Multi-Modality Activity Modeling. *International Joint Conference on Neural Networks (IJCNN)* (2018).

[5] Benish Fida, Daniele Bibbo, Ivan Bernabucci, Antonino Proto, Silvia Conforto, and Maurizio Schmid. 2015. Real time event-based segmentation to classify locomotion activities through a single inertial sensor. In *MobiHealth*.

[6] Kamran Ghasedi Dizaji, Xiaoqian Wang, and Heng Huang. 2018. Semi-supervised generative adversarial network for gene expression inference. In *The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.

[7] Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing* 25, 7 (2016).

[8] Travis R Goodwin and Sanda M Harabagiu. 2017. Deep Learning from EEG Reports for Inferring Underspecified Information. *AMIA Summits on Translational Science Proceedings* 2017 (2017).

[9] Haodong Guo, Ling Chen, Liangying Peng, and Gencai Chen. 2016. Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble. In *UbiComp*. ACM.

[10] Amir Harati, Meysam Golmohammadi, Silvia Lopez, Iyad Obeid, and Joseph Picone. 2015. Improved EEG event classification using differential energy. In *Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE.

[11] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM.

[12] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NIPS)*.

[13] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*.

[15] Oscar D Lara, Alfredo J Pérez, Miguel A Labrador, and José D Posada. 2012. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and mobile computing* 8, 5 (2012).

[16] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. 2016. Auxiliary deep generative models. *arXiv preprint:1602.05473* (2016).

[17] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).

[18] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[19] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* (2018).

[20] Siddharth Narayanaswamy, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. 2017. Learning disentangled representations with semi-supervised deep generative models. In *NIPS*.

[21] Iyad Obeid and Joseph Picone. 2016. The temple university hospital eeg data corpus. *Frontiers in neuroscience* 10 (2016).

[22] Augustus Odena. 2016. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583* (2016).

[23] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer.

[24] Lingxi Peng, Wenbin Chen, Wubai Zhou, Fufang Li, Jin Yang, and Jiandong Zhang. 2016. An immune-inspired semi-supervised algorithm for breast cancer diagnosis. *Computer methods and programs in biomedicine* 134 (2016).

[25] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012).

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*.

[27] Robin Tibor Schirrmeister, Lukas Gemein, Katharina Eggensperger, Frank Hutter, and Tonio Ball. 2017. Deep learning with convolutional neural networks for decoding and visualization of EEG pathology. *arXiv preprint:1708.08012* (2017).

[28] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder variational autoencoders. In *Advances in neural information processing systems (NIPS)*.

[29] Jost Tobias Springenberg. 2016. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *The International Conference on Learning Representations (ICLR)* (2016).

[30] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. 2016. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*. Springer.

[31] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*. Springer.

[32] Lina Yao, Feiping Nie, Quan Z Sheng, Tao Gu, Xue Li, and Sen Wang. 2016. Learning from less for better: semi-supervised activity recognition via shared structure discovery. In *UbiComp*. ACM.

[33] Xiang Zhang, Lina Yao, Chaoran Huang, Sen Wang, Mingkui Tan, Guodong Long, and Can Wang. 2018. Multi-modality Sensor Data Classification with Selective Attention. *IJCAI* (2018).

[34] Saeedeh Ziyabari, Vinit Shah, Meysam Golmohammadi, Iyad Obeid, and Joseph Picone. 2017. Objective evaluation metrics for automatic classification of EEG events. *arXiv preprint arXiv:1712.10107* (2017).