CPGAN: Curve Clustering Architecture Based on Projected Latent Vector of Generative Adversarial Network

AIYU ZHENG\textsuperscript{1}, JIANGHUI CAI\textsuperscript{1,2}, HAIFENG YANG\textsuperscript{1,2}, AND XUJUN ZHAO\textsuperscript{1}

\textsuperscript{1}School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China
\textsuperscript{2}Shanxi Key Laboratory of Advanced Control and Equipment Intelligence, Taiyuan 030024, China

Corresponding authors: Jianghui Cai (jianghui@tyust.edu.cn) and Haifeng Yang (hfyang@tyust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1931209 and Grant U1731126, in part by the Shanxi Province Key Research and Development Program under Grant 201803D121059 and Grant 201903D121116, and in part by the National Development and Reform Commission.

ABSTRACT Although Generative Adversarial Network (GAN) has obtained remarkable achievements in the image analysis and generation, its exploration in GAN-based curve clustering is still limited. The latent space of curve data is often used for clustering. However, the distance geometry in the latent space does not reflect the inherent clusters. In this paper, we propose CPGAN (Curve Clustering Architecture based on Projected Latent Vector of Generative Adversarial Network) for the clustering of curve dataset. Firstly, a novel GAN network structure, which utilizes a projector \( P \) composed of the transposed convolutional network to reconstruct the latent space of curve data, is proposed. CPGAN utilizes the concatenation of discrete code and Gaussian noise as a latent vector to preserve the implicit signal and structure of the cluster. Secondly, the loss function with two regularizations for CPGAN is proposed to guarantee the robustness and effectiveness of the model. Based on these, the jointly trained projector \( P \) is used to participate in the clustering process, while the generator can be used in the generating process. Finally, the spectral dataset from the LAMOST survey, the UCI dataset, and the UCR dataset are used as experimental data to evaluate clustering performance, the robustness of CPGAN, and further application on anomalous detection. CPGAN presents higher results than other methods.

INDEX TERMS Machine learning, generative adversarial network, curve data, clustering, anomalous detection.

I. INTRODUCTION

With more and more curve data (e.g. time-series data, temperature) are being recorded, correlative analysis technologies for such data are also thriving [1]. Curve data play an important role in the exploration of astronomy, medical, meteorological, geological exploration and other fields. The description of a curve data is usually a kind of one-dimensional digital signal which contains a large amount of data [2]–[4]. How can we automatically analysis and handling this kind of dataset? Can we even develop some standard templates for these datasets?

The analysis methods for curve dataset include clustering, regression, association rules, feature engineering, and classification [6], [9]. Clustering has been extensively applied in unsupervised learning with diverse approaches for mining curve dataset [7], [8] mentioned that the primary purposes of clustering have been to separate the original data into classes, and it is better if the clustering process started along with the low dimensional representation of the original real data. In order to effectively cluster the curve data, it is necessary to compress the curve data signal while keeping the shape of the curve unchanged, and then extract the curve features.

On the other hand, deep generative approaches drive another kind of unsupervised learning. Among them, the two most prominent methods are Variational Autoencoder (VAE) [5] and Generative Adversarial Network (GAN) [10]. Both of them obtain remarkable success in image generation, e.g. super-resolution tasks, semantic segmentation, etc. Variational Autoencoder (VAE) using an encoder network and a
decoder network to respectively accomplish data reduction and data reconstruction, and added the latent variables that obey Gaussian distribution to enhance the generalization ability of the model. Generative adversarial networks (GANs) jointly learn to generate synthetic data while learning a discriminator [10].

Based on these approaches, many kinds of researches are trying to combine clustering approaches and generative models to obtain better clustering effects. Cluster-GAN [11] designs a GAN training methodology that can cluster and generate images in the latent space by introducing a projection module in the network structure. ClusterGAN utilizes the affinity matrix generated by the subspace self-expressiveness to intensify the representation of latent space. [12] proposes an unconditional generative adversarial model called K-Means-GAN(KM-GAN), which incorporates the idea of updating centres in k-means into GANs. According to our exploration and investigation, few of the previously proposed GAN based models are designed for curve clustering [11]. Besides, However, the distribution of direct projection of original data should look similar to the latent space distribution.

A. MOTIVATIONS
Most exploration of clustering based on generative model mainly focus on the improvement of the representation of latent space. The latent space of VAE and GANs not only provides dimensionality reduction but also gives rise to a novel representation of the original real data. The motivations of this work can be summarized as follows:

1) The direct clustering using the recognition model of GAN does not always result in identifying object category-level information [24]. The combination of traditional unsupervised learning methods and GANs might produce a better result than direct clustering.
2) Most of previously proposed GAN based clustering models are designed for two-dimensional datasets [12]. It is necessary to design corresponding models for curve datasets.
3) Using lower-dimensional feature representation of the original data as input to the clustering algorithm can produce a better result. However, the GAN latent space does not maintain cluster structure 11]. This means that the latent space generated by the traditional GAN structure cannot be directly used for clustering.

B. CONTRIBUTIONS
This paper proposes a novel curve clustering architecture based on the projected latent vector of generative adversarial network named CPGAN, which can generate curves and clustering by using the latent space. The overall structure of CPGAN is shown in Figure 1. In order to obtain more accurate projection effects, we introduce projector $P$ [11] into our network structure and modify the objective function of GAN. This model additionally provides explainable results, which can assist in cluster datasets. CPGAN can produce robust reconstructed data because of the utilization of regularization.

In summary, our main contributions are as follows:

1) This paper proposes a convolutional generative adversarial network architecture, which fulfills the clustering task for curve data based on the concept of a two-stage clustering strategy. Based on the ClusterGAN structure, the network is improved by using 1-dimensional convolutional neural network instead of fully connected network and modifying the original two regularization parameters as one parameter, making it more suitable for processing curve data. The GAN model of this architecture can project the original data space to the clustering-specific latent space according to an independent projector $P$.

2) The latent variables composed of discrete code(one-hot) and continuous code are used as the representation of raw curve data to participate in the second stage of the clustering task. These latent variables preserve expressible information of real clusters and break the smoothness of the latent space. Therefore, the clustering for such latent variables with good expressiveness will obtain better results.

3) In order to avoid the overfitting of reconstruction which might lead to low clustering accuracy. This proposes a clustering-specific loss function with regularizations. The robustness of adversarial regularization is well expressed.

4) Experiments on the LAMOST [13] dataset, the UCI dataset, and the UCR dataset shows that the proposed method achieves nicer results than other GAN/VAE-based methods and other methods on the aspect of clustering performance, robustness of the model, and further application on anomalous detection.
The rest of this paper is organized as follows. Section 2 introduces the related works of the combination of GANs and clustering. The network structure, regularizations and CPGAN architecture are introduced in Section 3. The clustering performance, discussions of robustness and application on anomalous detection are presented as experiments in Section 4. Our work is concluded in Section 5.

II. RELATED WORK

With the development of the variant of the autoencoder(stacked denoising autoencoders [14], sparse autoencoder [15] and deep CCA [16]), deep learning methods have been widely used for dimensionality reduction. Meanwhile, the encoder-decoder structure helps to build the architecture for deep unsupervised subspace clustering [17]. Recently, more and more unsupervised learning is driven by deep generative approaches, the most prominent being Generative Adversarial Network(GAN). GANs have better performance than autoencoder in generating high fidelity samples [11]. It makes the strong latent representations of GAN producing improved clustering results be possible.

Most of the recent studies GANs focus on the image generation and its associated subject area, e.g. anomaly detection, imputation [18]. StyleGAN [19] proposes an alternative generator architecture for GAN, borrowing from style transfer literature. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and disentangles the latent variable. BeatsGAN [20] combines the idea of AE and GAN, proposes a reconstruction-based method for detecting anomalous curve. It utilizes deep autoencoder, and GAN’s reconstruct loss to guarantee the ability of representativeness of latent code. In addition, BeatsGAN introduces adversarial regularization in its training process. MSGAN [21] devises two generators to generate the samples containing spatial(curve) and spectral information, respectively, and the discriminator is devised to extract joint spatial-spectral features and output multiclass probabilities.

Another kind of GANs aims at finding out the implicit information between latent space and original data, and even attempt to integrate the concept of clustering approaches. InfoGAN [22] employ discrete latent variables to create the interpretable and disentangled latent variables. [23] fuse the features learned by adversarial training with a traditional unsupervised learning approach, k-means clustering, and show that this combination produces better results than direct prediction. CatGAN [24] combines neural network classifiers with an adversarial generative model that regularizes a discriminatively trained classifier, and the generator which learned alongside the classifier can generate high visual fidelity image. KM-GAN [12] proposes an un-conditional generative adversarial model, which incorporates the idea of updating centres in K-Means on the features extracted from the discriminator. ClusterGAN utilizes a mixture of discrete and continuous latent variables, and it proposes a novel backpropagation algorithm accommodating the discrete-continuous mixture, as well as an explicit inverse mapping network to obtain the latent variables. ClusterGAN claims it is the first work that addresses the problem of clustering in latent space. Additionally, to the best of our knowledge, none of the recent work is to solve the issue of clustering in latent space for curves. Hence, we promote CPGAN to accomplish the clustering task for curve data.

III. PROPOSED METHOD

In this section, the GAN structure, regularizations and architecture of the proposed method(CPGAN) is introduced. At first, the problem is set as follows:

Let \( X \in \mathbb{R}^{M \times N} \) be a set of curve data, which has \( N \) instances, and \( M \) dimensions. We consider the problem of learning a model from \( X \), such that model transforms the data into a representative latent vector with category information. Suppose that there are \( C \) classes in \( X \), so the objective is to cluster the latent space mentioned above into \( C \) categories.

A. MODIFIED NETWORK STRUCTURE

[23] has revealed that using a traditional clustering algorithm, i.e., k-means, turns out to be much more effective at grouping data from similar categories than the approach of directly predicting the categorical groups. Therefore, this problem can naturally be considered as a two-stage clustering task. The first stage is to learn a model that can project the original data space into low-dimensional latent space by using GAN structure. This resulting model is used as a preprocessing procedure in the second stage. The second stage is to complete the clustering task on the latent space based on the previous work and output the category results by using k-means. Besides, the generator of GAN can be used to generate curve data for imputation and abnormal detection. The GAN structure of our method is shown in Figure 1.

The GAN structure of our method consists of three components, except for the same generator \( G \) and discriminator \( D \) as vanilla GAN, we introduce a projector \( P \) to fulfill the mapping of raw data space to latent space.

These three are implemented as neural networks parameterized by \( \Theta_G \), \( \Theta_D \) and \( \Theta_P \) respectively. DCGAN [25] constructs the networks using CNN with Conv-Deconv layers, and it performs well in image feature extraction. And there are many studies show that CNN networks are more robust than LSTMs for curve [26]. Therefore, CPGAN uses 1-D(dimentional) convolutional neural and 1-D transpose convolution neural to compose the entire model.

As for the detailed design of the network, with the concept of DCGAN, we use a fully convolutional neural network instead of using spatial pooling. This allows the network to learn more suitable result for spatial downsampling. Moreover, avoid using too many fully connected layers after the convolution layer, because the fully connected layer increases the stability of the model, but also slows down the convergence rate. The network of CPGAN uses the ReLU or LeakyReLU activation function, and Batch Normalization (BN) is used in some places.
The input of the generator is composed of Gaussian noise and one-hot coding. First, the fully connected layer is used for feature expansion, and then the deep convolution operation is performed after a Reshape. The discriminator accepts the curve input and then performs transposed convolution. The last convolution layer of the discriminator is generally flattened, then passes through the fully connected layer, and finally outputs the result by a single connected layer. The projector uses a fully connected layer with a size equal to the input length of the generator to obtain projected results. This layer is segmented into two parts, the tail of the layer and the input length of the generator to obtain projected results. Finally, outputs the result by a single connected layer. The main idea is to project the generations into the latent space and one-hot coding. First, the fully connected layer is used to map the input to the latent space distribution. As for the activation function, we use the tanh function.

The generator can also be considered to be a projection from latent space $Z \in \mathbb{R}^{Q \times N}$ to the data space $X$, where $Q < M$. As shown in the experimental results of [24] and [11], the Gaussian components of latent space in vanilla GAN tend to crowd and become redundant. Thus vanilla GAN cannot cluster well in the latent space. We introduce categorical variables (one-hot code) to solve the problem effectively. Therefore, $Z$ is sampled from a prior that consists of normal random variables cascaded with one-hot encoded vectors. That means $z = (z_n, z_c)$, where $z_n \sim N$ and $z_c$ is the one-hot code randomly sampled from $\mathcal{U}(1, 2, \ldots, C)$ distribution. As for the $\sigma$ of the $N$, small variances $\sigma$ are chosen to guarantee the clusters in latent space $Z$ are separated (According to the recommendation of [11] $\sigma = 0.10$ is appropriate). One-hot code can make a proper impact on GAN’s training, and make $G$ only generate corresponding curve data $X'$. In the generator $G$, after the fully connected layer and reshape layer, 1-D TCN layers are stacked layer by layer along with the ReLU activation function and batch normalization strategy except the last layer (which utilizes tanh function).

The discriminator is designed to be a projection from the real data space $X$ to the binary value 0/1, which corresponds to the probability of real or fake. The vanilla GAN uses a two-player game between $G$ and $D$, which is defined by the min-max objective:

$$\min_{\Theta_G, \max_{\Theta_D}} \mathbb{P}_X q(D(x)) + \mathbb{P}_Z q(1 - D(G(z)))$$ (1)

where $\mathbb{P}_X$ is the distribution of $X$, $\mathbb{P}_Z$ is the distribution of $Z$ on the latent space and $q(\cdot)$ is the quality function (in vanilla GAN, $q = \log \pi$). Then jointly updating $\Theta_G$ and $\Theta_D$ to make the distribution of $X$ and $X'$ more and more similar. Obviously from a certain perspective, $D$ can be regarded as the inverse operation of $G$, that is, a kind of downsampling [25]. This means the category information can be extracted from the downsampling procedure [22], [25]. However, this kind of method does not act directly on the latent space which participates in the clustering process. We need a network that is specifically targeted at the reconstruction of latent space to complete the clustering task. In discriminator $D$ of CPGAN, we use 1-D CN to construct its network.

Different from vanilla GAN, inspired by [11], CPGAN has a projector $P$, a convolutional neural network parameterized by $\Theta_P$, which is used to project real data space $X$ and generated data space $X'$ to the latent space $Z$. $P$ directly participates in the min-max objective as a clustering specific loss term. An independent mapping mechanism avoids discriminator-based parameter sharing, which can get more generalized models and adapt to more complex data sets. Due to the previous exploration [27], the reconstruction of latent vector $z$ and $z'$ often accomplished by regularization to guarantee the excellent effects. Therefore, we revise the GAN objective function as the following form:

$$\min_{\Theta_G, \Theta_P} \max_{\Theta_D} \mathbb{P}_X q(D(x)) + \mathbb{P}_Z q(1 - D(G(z))) + \lambda (E_{z \sim \mathbb{P}_Z} \| z_n - P(G(z_n)) \|^2_2 + E_{z \sim \mathbb{P}_Z} \mathcal{H}(z_c, P(G(z_c))))$$ (2)

where $\mathcal{H}(\cdot)\ldots$ is the cross-entropy loss, and the $\|\cdot\|^2_2$ is the mean-square error. $\lambda$ is the regularization coefficient, which decides the importance of the portion of the latent variables. This coefficient is a union of the coefficients $\beta_n$ and $\beta_c$. The regularization learns to project $P(G(z))$ to the centroid of the respective cluster (i.e. the $\mu$ of the latent space distribution) by updating their parameters iteratively. This is similar to the concept of K-Means.

The GAN training is implemented by jointly training $P, G$ and $D$ to obtain appropriate parameters $\Theta_P, \Theta_G$ and $\Theta_D$.

**B. REGULARIZATIONS FOR CLUSTERING**

In order to avoid overfitting, diminished gradient and mode collapse, we design two regularizations. The first one is to minimize the reconstructed loss of the projected latent vectors which come from original data $x$ and generated data $x'$:

$$L_{pfo} = \| P(x) - P(x') \|^2_2$$ (3)

Thus the loss function for projector $P$ is:

$$L_P = \lambda_1 (E_{z \sim \mathbb{P}_Z} \| z_n - P(G(z_n)) \|^2_2 + P(G(z_c)))) + \lambda_2 E_{x \sim \mathbb{P}_X, x' \sim \mathbb{P}_X'} \| P(x) - P(x') \|^2_2$$ (4)

where $\lambda_1$ and $\lambda_2$ are the weighting parameters to adjust the influence of these regularizations. On the one hand, $P$ aims at minimizing the difference between projected $z' = (P(G(z_n)), P(G(z_c)))$ and $z \sim \mathbb{P}$. On the other hand, it also attempts to reduce the difference between the two mapped latent vectors which come from original data and generated data. Besides, for the $q(\cdot)$ mentioned, we use $q(x) = x$ (WGAN-GP [28]).

As shown in the lower right of Figure 1, the other regularization is the pairwise feature matching loss which minimizes differences of the statistics between original and generated curve data, learned in hidden layers of the discriminator $D(\cdot)$ [20]:

$$L_{pffm} = \| f_D(x) - f_D(x') \|^2_2$$ (5)

where $f_D(\cdot)$ is the activated output vector of the fully connected layer of the discriminator. Therefore, the loss function...
for generator $G$ is:

$$L_G = E_{z \sim P_z} q(D(G(z))) + \lambda_1 E_{z \sim P_z} q(z)$$

where $\lambda_1$ uses same set with Eq(2), $\lambda_3$ is the parameter for controlling impact of this regularizations, and $x' = G(z)$. Meanwhile, the loss function of the discriminator is to minimize the equation below:

$$L_D = E_{x \sim P_x} q(D(x)) + E_{z \sim P_z} q(1 - D(G(z)))$$

The total GAN training approach summarized as Algorithm 1. Adam [20] is used to optimize the parameters during the Backprop procedure.

C. CPGAN ARCHITECTURE

The overall architecture of our method using spectra data as an example is illustrated in Figure 2. It is divided into three parts. The first part is mainly used to train the GAN model. At the beginning of this part, the original spectra data are preprocessed and shuffled by tailoring and normalizing to the training set. Then this dataset is pushed into GAN structure to obtain an appropriate model by using Algorithm 1.

After that, we can get a well-trained projector $P$ and generator $G$. For the second part(clustering task), we can put the preprocessed test data set into the projector $P$ to obtain the representative latent vectors. These latent vectors preserve important clustering information of the original data, and it is sufficient to represent the original data to participate in the clustering process. Then the latent vectors are pushed into the clustering method to produce category results. As for the clustering method, according to the advice of [11], we utilize k-means to cluster these latent vectors. In the following section, we will investigate the influence of different mainstream clustering methods on the same dataset for CPGAN. Formally, Algorithm 2 summarizes the approach.

For the third part(generating task), the trained generator $G$ can be used to generate corresponding spectra data by manipulating the latent code $z_c$ and noise $z_n$. By adjusting the value of $z_c$, $G$ can produce different categories of the spectrum. The abundant experiments have demonstrated $G$ can generate high-quality spectra data.
Algorithm 2 CPGAN

Input: GAN structure $G_S$, dataset $X$, number of clusters $C$
Output: Cluster set $CS$
1: Obtain train set $X_{train}$ and test set $X_{test}$ through pre-processing $X$
2: Train $G_S$ model with $X_{train}$ and $C$ by using Algorithm 1
3: Obtain latent vectors $L_{test}$ using the projector $P$ of $G_S$ with $X_{test}$; $L_{test} = P(X_{test})$
4: Initialize the parameter $k$ of k-means using $C$
5: Cluster $L_{test}$ by k-means: $CS = k$-means($L_{test}$)
6: return $CS$

IV. EXPERIMENT AND ANALYSIS
A. DATASETS AND COMPARISON METHODS

1) LARGE SKY AREA MULTI-OBJECT FIBER SPECTROSCOPIC TELESCOPE (LAMOST) [13]: The spectra data from LAMOST on optical bands provide nebula, galaxy and stars source to research as the curve. LAMOST is available at http://dr5.lamost.org. Each instance of LAMOST has 5 dimensions, and each dimension has 3908 wavelengths. In our experiments, the data are preprocessed before application in the method by the following steps: Firstly, we have randomly sampled 6000 spectra data from January to May 2017 (including four categories, respectively F0, M0, QSOs and WD, each category has 1500 instances) from LAMOST as the original dataset. Secondly, we choose the fifth dimension of each spectrum, which represents the wavelength flux curve (the flux equivalent to the longitudinal coordinate of the two-dimensional curve) as the input data. Thirdly, for each spectrum, the top 308 fluxes of the spectrum are cut due to the low signal-noise ratio. In order to reduce the signal redundancy, we sequentially sample one flux for every three fluxes for the rest fluxes. Finally, the dataset is normalized to a common scale $[-1, 1]$.

2) CHARACTER TRAJECTORIES (CT) [34]: The data consists of 2858 character samples. It is available at http://timeseries classification.com. Among them, the size of the train set is 1422, and the size of the test set is 1436. The data is normalised. Each instance is a 1-dimensional curve and its length is 512. ShapesAll has 60 different shapes of curves. Consider the number of instances of this dataset is very small, we supplemented the dataset to 3600 by Gaussian distribution (each instance generates two new instances with $\mu = 0, \sigma = 0.1$). Each original instance can generate two new synthetic instances by adding such Gaussian noise.

The comparison methods are:

- InfoGAN [22]: An information-theoretic extension to the Generative Adversarial Network that is able to learn disentangled representations in a completely unsupervised manner.
- ClusterGAN [11]: By sampling latent variables from a mixture of one-hot encoded variables and continuous latent variables, coupled with an inverse network trained jointly with a clustering specific loss to achieve clustering in the latent space.
- BeatsGAN [20]: BeatGAN outputs explainable results to pinpoint the anomalous time ticks of an input beat, by comparing them to adversarially generated beats.
- KM-GAN [12]: An unconditional generative adversarial model, called K-Means-GAN(KM-GAN), which incorporates the idea of updating centres in K-Means into GANs.
- StyleGAN [19]: An alternative generator architecture for generative adversarial networks, borrowing from style transfer literature.
- CVAE [46]: A scalable deep conditional generative model for structured output variables using Gaussian latent variables.

Implementation Details: The whole networks are trained by ADAM algorithm, learning rate $= 0.0001$. All the comparison methods’ parameters use the recommendation of their papers, and some neurons of some network structure are changed to 1-D version. The implementation of our method is based on the characteristics of the dataset, multiple evaluations and the practice in the citations. Different from the ClusterGAN uses fully connected layer to deal time-series data in its original paper, CPGAN uses one-dimensional transposed convolutional (1D-TCN) neural and convolutional neural (1D-CN) conforming the concept of DCGAN to apply to all datasets (the being compared GAN based methods in the following experiments also uses these setting). Moreover, the GAN structure design follows the previous description in Modified Network Structure Section. Inspired by [20] and actual experimental effects, we use 4*1 kernel size, 2*1 stride. The kernel numbers of every layer for every dataset and other detailed information are shown in Table 1, Table 2 and Table 3 (e.g. “FC 1024 LReLU; 1D-TCN 64,4*1, 2*1 ReLU BN” means “fully connected layer, layer size: 1024, activation: Leaky ReLU; one-dimensional transposed convolutional layer, kernel number: 64, kernel size: 4*1, stride: 2*1, activation: ReLU, Batch Normalization”). All the kernel sizes are designed according to the size of each instance in each dataset, the layer structure is also designed in this way (fully connected layer is used to expand and reduce...
A. Zheng et al.: CPGAN: Curve Clustering Architecture Based on Projected Latent Vector of GAN

### TABLE 1. Neural network structure for LAMOST (TCN-transposed convolutional neural, CN-convolutional neural).

| Projector | Generator | Discriminator |
|-----------|-----------|---------------|
| Input $X \in \mathbb{R}^{128 \times 2}$ | $z_n(158) + z_c(5)$ | $z_n(158) + z_c(5)$ |
| 1D-CN 32, 4*1, 2*1 LReLU | FC 1024 ReLU BN | ID-CN 32, 4*1, 2*1 LReLU |
| 1D-CN 64, 4*1, 2*1 LReLU | FC 2048 ReLU BN | ID-CN 64, 4*1, 2*1 LReLU |
| 1D-CN 128, 4*1, 2*1 LReLU | FC 75*256 ReLU BN Reshape | ID-CN 128, 4*1, 2*1 LReLU BN |
| 1D-CN 256, 4*1, 2*1 LReLU | ID-TCN 64, 4*1, 2*1 LReLU BN | ID-TCN 64, 4*1, 2*1 LReLU BN |
| FC 2048 LReLU | ID-TCN 32, 4*1, 2*1 LReLU BN | ID-TCN 32, 4*1, 2*1 LReLU BN |
| FC 1024 LReLU | FC 1024 LReLU | FC 1024 LReLU |
| FC 163 for $z'$, Softmax on last 5 of $z'$ for $z'_c$ | | |

### TABLE 2. Neural network structure for CT (TCN-transposed convolutional neural, CN-convolutional neural).

| Projector | Generator | Discriminator |
|-----------|-----------|---------------|
| Input $X \in \mathbb{R}^{192 \times 2}$ | $z_n(100) + z_c(20)$ | $z_n(100) + z_c(20)$ |
| 1D-CN 64, 4*1, 2*1 LReLU | FC 1024 ReLU BN | ID-CN 64, 4*1, 2*1 LReLU |
| 1D-CN 128, 4*1, 2*1 LReLU | FC 128*46 ReLU BN Reshape | ID-CN 128, 4*1, 2*1 LReLU BN |
| FC 1024 LReLU | ID-TCN 64, 4*1, 2*1 LReLU BN | ID-TCN 64, 4*1, 2*1 LReLU BN |
| FC 120 for $z'$, Softmax on last 20 of $z'$ for $z'_c$ | ID-TCN 1, 4*1, 2*1 tanh | ID-TCN 1, 4*1, 2*1 tanh |
| | FC 1024 LReLU | FC 1024 LReLU |

### TABLE 3. Neural network structure for ShapesAll (TCN-transposed convolutional neural, CN-convolutional neural).

| Projector | Generator | Discriminator |
|-----------|-----------|---------------|
| Input $X \in \mathbb{R}^{128 \times 2}$ | $z_n(128) + z_c(60)$ | $z_n(128) + z_c(60)$ |
| 1D-CN 32, 4*1, 2*1 LReLU | FC 1024 ReLU BN | ID-CN 32, 4*1, 2*1 LReLU |
| 1D-CN 64, 4*1, 2*1 LReLU | FC 128*64 ReLU BN Reshape | ID-CN 64, 4*1, 2*1 LReLU BN |
| 1D-CN 128, 4*1, 2*1 LReLU | ID-TCN 64, 4*1, 2*1 LReLU BN | ID-TCN 64, 4*1, 2*1 LReLU BN |
| FC 1024 LReLU | ID-TCN 32, 4*1, 2*1 LReLU | ID-TCN 32, 4*1, 2*1 LReLU |
| FC 188 for $z'$, Softmax on last 60 of $z'$ for $z'_c$ | ID-TCN 1, 4*1, 2*1 tanh | ID-TCN 1, 4*1, 2*1 tanh |
| | FC 1024 LReLU | FC 1024 LReLU |

### B. CLUSTERING PERFORMANCE

1) EVALUATION OF GAN

The Frechet Inception Distance score, or FID for short, is a metric that calculates the distance between feature vectors calculated for real and generated data [29]. It is used to evaluate the quality of generated samples from a GAN. We record the FID values of different GAN-based methods in Figure 3. The lower the FID is, the better quality the model has. Figure 3 illustrates CPGAN which has the shortest rectangle achieves the best sample quality compared with other methods with all datasets.

We run CPGAN on LAMOST to investigate its generative ability. As shown in Figure 4, the first two rows are generated spectra data, which have 4 categories (named M0, F0, WD, QSOs), and the third row is corresponding original preprocessed spectra data. Obviously, compared with original data, generated spectra data present high-quality spectral curve characteristics. The position of the absorption line and the emission line is correct. Meanwhile, the trend of each generated spectral curve is also consistent with its corresponding original data, and the curves of different categories also show obvious differences. It is difficult to discriminate generated data from real data.

CPGAN utilizes three real datasets, including a LAMOST spectral dataset, a UCI dataset, and a UCR dataset to evaluate itself. CPGAN uses a two-stages strategy to accomplish complex clustering goals, and the training of the GAN model needs a large number of data. The following three datasets have high dimensional instances, large data volume and complex curve shape, which are very suitable for CPGAN evaluation.

On the other hand, we also provide the generated results for the CT dataset. In Figure 5, the left part is the real data and the right part is the generated data. The generated character...
FIGURE 4. Generated spectra (normalized) from CPGAN and real spectra (normalized): LAMOST.

FIGURE 5. Generated curve data from CPGAN and real curve data: CT.

curve has the correct shape, and its handwriting style is well maintained. CPGAN shows good generating ability.

2) ACCURACY OF CLUSTERING

Accuracy and validation of clustering are important goals of our works. We compared these indicators of CPGAN with other possible GAN&VAE based clustering approaches we conceive. We also add the comparison of traditional clusterings such as Non-negative Matrix Factorization (NMF) [30], Agglomerative Clustering (AGGLO) [31], [32] and Spectral Clustering (SC) [8], [34]. AGGLO uses Euclidean distance and ward linkage strategy. NMF used KL-divergence loss and initialized with SVD. The kernel of SC is RBF. As for the indicators, normalized mutual information (NMI) [37], Adjusted Rand index (ARI) [38], and clustering purity (ACC) [39] is used as a standard of measurement. The results are shown in Figure 6.

Figure 6(a) illustrates the ACC value of the comparison algorithms with two datasets. CPGAN reaches the highest ACC values on both datasets (LAMOST-0.83, CT-0.84, ShapesAll-0.77). The value of other GAN-based methods are close to CPGAN, especially StyleGAN (LAMOST-0.82, CT-0.83, ShapesAll-0.75) and ClusterGAN (LAMOST-0.81, CT-0.82, ShapesAll-0.76). This is because these methods correctly extract the inherent feature information of clusters in latent space. Another GAN-based method like InfoGAN cannot produce nice ACC values. But CV AE has shown better results than InfoGAN. This should be due to its inference network which can better preserve cluster structure while mapping original data to latent variables. Obviously, directly use traditional clustering methods (NMF, AGGLD, SC) on the datasets result in terrible results. This also proves that the extraction of raw data is necessary before clustering.

Figure 6(b) illustrates the NMI value of the comparison algorithms with two datasets. CPGAN achieves nice performance on both datasets again (LAMOST-0.82, CT-0.80, ShapesAll-0.75). ClusterGAN and StyleGAN also present nice performance on the CT dataset. On the contrary, traditional clustering methods still show bad results on these two datasets.

Figure 6(c) illustrates the ARI value of the comparison algorithms with two datasets. CPGAN still shows nice results on these three datasets.

Through the above experiments, we find that overall the indicators of CPGAN are better than ClusterGAN. This is
FIGURE 6. The clustering metrics results of CPGAN and other methods on three datasets. The closer the value is to 1, the better the results.

because the regular terms used by CPGAN improve the generalization ability of the model, thus providing better clustering results.

C. DISCUSSIONS OF ROBUSTNESS

This section focuses on the robustness of CPGAN. Mainly answered the following three questions about CPGAN’s clustering effects:

1) In the previous experiments, CPGAN uses k-means to cluster the latent space. Does CPGAN use different clustering algorithms will affect clustering performance?

2) Cluster number \( c \) as an important parameter of CPGAN, it is defined according to the number of real clusters in the previous experiments. What if it is varied? Does this affect the accuracy of clustering to a large extent? And how about the large scale number of a cluster?

3) CPGAN uses two regularizations. How does changing the parameters of the regularizations affect the clustering effects?

1) CLUSTERING METHODS ROBUSTNESS

The purpose of this experiment is to evaluate the effect of using different clustering algorithms on CPGAN’s clustering results. CPGAN achieves the goal of clustering (as shown in Figure 2-“Clustering”) by a clustering method. In the previous experiments, we select k-means to achieve the clustering process. Now we utilize other clustering methods to run CPGAN to investigate the robustness of the produced latent vector. The clustering method is one of the influential factors of our method. Whether choosing different clustering algorithms will affect clustering results? And what if we directly use latent variables to participate in clustering?

We compare different clustering methods (include KMeans used in the above experiments, DBSCAN, Mean-Shift(MS), Spectral Clustering (SC), and Single-linkage clustering(SLC)) and the situation without the clustering method(WC) on CT dataset by using ACC. The result shown in Figure 7 illustrates that the ACC is not severely affected by the selected clustering algorithm. The value is still within the acceptable range(for LAMOST, ACC value range from 0.8180 to 0.8284, for CT, ACC value range from 0.8317 to 0.8426). This is because CPGAN already provides excellent informational latent vectors by utilizing one-hot codes. The distance geometry in the latent space can reflect the inherent clusters. Thus, the influence of the clustering methods selection is reduced to a reasonable range. Besides, WC shows bad performance. Its ACC value is significantly lower than the case with the clustering method. This is because although the discrete latent variables generated by the trained GAN model contain inherent cluster information, the interpolation of the model still has led to some variability in the continuous latent variables, which eventually led to this result.

2) INFLUENCE OF CLUSTER NUMBER

We also investigate the influence of cluster number \( C \) for the second question. It is answered by two experiments.

In the first experiment, we use the CT dataset to exam the influence of varied cluster number \( C \). The results are shown in Table 4. The number of clusters is initialized different values around the number of real categories. The bold items
TABLE 4. ACC value of CPGAN result with different cluster number C on CT dataset.

| C  | 15 | 18 | 20 | 22 | 25 |
|----|----|----|----|----|----|
| ACC| 0.57| 0.72| 0.84| 0.79| 0.68|

in the table are the number of real clusters. When the number of clusters C up form 20 to 25, ACC down from 0.84 to 0.68. When the number of clusters C down form 20 to 15, ACC down from 0.84 to 0.57. The farther C is from the correct cluster number, the worse the ACC is. But higher values of C lead to not severe impacts, while lower values of C opposite.

In the second experiment, we use the ShapesAll dataset to examine the influence of a large number of cluster C. ShapesAll has 60 categories, and 512 points in each instance. CPGAN is trained by the train set, and then clusters test set. As shown in Figure 8, the left part is train data(train set), and the right part is clustered data(test set). Obviously, the clustered data are very similar between each other. The clustering results show CPGAN produces nice performance even with such a high number of C.

3) REGULARIZATIONS ROBUSTNESS
In this experiment, we investigate the regularizations robustness of CPGAN for the last question. In the experiment,
TABLE 5. ACC value of CPGAN result with or without regularizations on CT dataset.

| Regularizations | ACC value (0) |
|-----------------|---------------|
| $\lambda_2 = 0, \lambda_3 = 0$ | 0.8241 |
| $\lambda_2 \neq 0, \lambda_3 \neq 0$ | 0.8406 |
| $\lambda_2 = 0, \lambda_3 = 0$ | 0.8297 |

TABLE 6. Results of Anomalous Detection on ShapesAll dataset.

| Method | AUC | AP |
|--------|-----|----|
| PCA    | 0.82 | 0.77 |
| BeatsGAN | 0.93 | 0.91 |
| VAE    | 0.88 | 0.86 |
| CPGAN  | 0.91 | 0.88 |

we control the regularization module to try to get the actual impact of regularization on the model. Metrics still use ACC.

As shown in the Table 5, we run CPGAN on CT dataset with 3 situations:

1) The case without regularizations($\lambda_2 = 0, \lambda_3 = 0$).
2) The case with two regularizations($\lambda_2 \neq 0, \lambda_3 \neq 0$).
3) The case with only one regularization($\lambda_2 = 0$ or $\lambda_3 = 0$).

Obviously, the accuracy drops without regularizations which also indicates our method has better robustness than ClusterGAN. But in situation 3, the removal of $\lambda_2$ leads to lower ACC compared with the removal of $\lambda_3$. This is because $\lambda_2$ directly impacts on the mapping process, but $\lambda_3$ more focuses on the generating process.

**D. APPLICATION ON ANOMALOUS DETECTION**

In this experiment, we use two metrics, AUC (Area Under ROC Curve) [40] and AP (Average Precision) [41], [42] to evaluate the accuracy of anomalous detection based on CPGAN. Here we use the ShapesAll dataset to fulfill this experiment. Top 50 categories of ShapesAll are considered as normal data; the rest are considered as anomalous data. The k-means is used as the clustering method in CPGAN, and the abnormal threshold is set as the sum of the average distance of each cluster and 1.5 times the standard deviation. The Principal Component Analysis (PCA) [43], [44], BeatsGAN, and VAE-based anomaly detection [45] are used as a comparison in this experiment.

As shown in Table 6, CPGAN has significantly better performance than PCA, but is slightly weaker than BeatsGAN. This is because GAN-based and VAE-based methods naturally have advantages on the complex dataset than linear models. This result proves that CPGAN has certain application capabilities.

**V. CONCLUSION**

In this paper, we propose a curve clustering architecture CPGAN for the clustering of curve dataset and the generation of curves. CPGAN uses a projector $P$ which composed of the transposed convolutional network to reconstruct the latent vectors to represent raw curve data. These latent vectors use discrete code to preserve the implicit signal and structure of the cluster. With the help of two regularizations in loss functions, the robustness and effectiveness of the clustering results are guaranteed. Based on these, the whole network of CPGAN is trained jointly to obtain appropriate model parameters. Then the trained models are used to participate in the clustering process and the generating process. Furthermore, the comparison with other clustering methods by using the LAMOST dataset, several UCI datasets and a UCR dataset illustrates that CPGAN is competent for clustering tasks and has a nice application ability.

**ACKNOWLEDGMENT**

The Guo Shou Jing Telescope (the Large Sky Area Multi-Object Fiber Spectroscopic Telescope, LAMOST) is a National Major Scientific Project built by the Chinese Academy of Sciences. LAMOST is operated and managed by National Astronomical Observatories, Chinese Academy of Sciences.

**REFERENCES**

[1] C. Abraham, P. A. Cornillon, E. Matzner-Lober, and N. Molinari, “Unsupervised curve clustering using B-Splines,” *Scandin. J. Statist.*, vol. 30, no. 3, pp. 581–595, Sep. 2003.

[2] D. J. Hand, G. J. McLachlan, and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*, vol. 38. New York, NY, USA: M. Dekker, 1998.

[3] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, vol. 20. Philadelphia, PA, USA: SIAM, 2007.

[4] L. M. Sangalli, P. Secchi, S. Vantini, and V. Vitelli, “K-mean alignment for curve clustering,” *Comput. Statist. Data Anal.*, vol. 54, no. 5, pp. 1219–1233, May 2010.

[5] D. P Kingma and M. Welling, “Auto-encoding variational bayes,” 2013, arXiv:1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114

[6] C. Qu, H. Yang, J. Cai, J. Zhang, and Y. Zhou, “DoPS: A double-peaked profiles search method based on the RS and SVM,” *IEEE Access*, vol. 7, pp. 106139–106154, 2019.

[7] Y. Li, J. Cai, H. Yang, J. Zhang, and X. Zhao, “A novel algorithm for initial cluster center selection,” *IEEE Access*, vol. 7, pp. 74683–74693, 2019.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[9] Y. Yang, J. Cai, H. Yang, J. Zhang, and X. Zhao, “TAD: A trajectory clustering algorithm based on spatial-temporal density analysis,” *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112346.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, and D. Warde-Farley, “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680.

[11] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “ClusterGAN: Latent space clustering in generative adversarial networks,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 4610–4617.

[12] C. Wang, Z. Chen, K. Shang, and H. Wu, “Label-removed generative adversarial networks incorporating with K-Means,” *Neurocomputing*, vol. 361, pp. 126–136, Oct. 2019.

[13] A. L. Luo, Y. H. Zhao, G. Zhao, L. C. Deng, Y. W. Liu, Y. P. Jing, and Y. Q. Chu, “The first data release (DR1) of the LAMOST regular survey,” *Res. Astron. Astrophys.*, vol. 15, no. 8, p. 1095, 2015.

[14] S. Xiang, F. Nie, and C. Zhang, “Learning a mahalanobis distance metric for data clustering and classification,” *Pattern Recognit.*, vol. 41, no. 12, pp. 3600–3612, Dec. 2008.

[15] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Jun. 2011, pp. 215–223.

[16] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, “Deep canonical correlation analysis,” in *Proc. Int. Conf. Mach. Learn.*, Feb. 2013, pp. 1247–1255.

[17] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, “Deep subspace clustering networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 24–33.

[18] Y. Luo, X. Cai, Y. Zhang, and J. Xu, “Multivariate time series imputation with generative adversarial networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1596–1607.
AIYU ZHENG is currently the Chief Assistant of computer application technology with the Taiyuan University of Science and Technology, Taiyuan, China. His current research interests include data mining and artificial intelligence. He is a member of China Computer Federation (CCF).

JIAHUI CAI is currently the Chief Professor of computer application technology with the Taiyuan University of Science and Technology, Taiyuan, China. He is a long-term member of the Institute for Intelligent Information and Data Mining. His research concerns the data mining and machine learning methods in specific backgrounds of astronomical informatics, seismology, and mechanical engineering. He is a Senior Member of China Computer Federation (CCF).

HAIFENG YANG is currently a Professor of computer application technology with the Taiyuan University of Science and Technology, Taiyuan, China. He is a long-term member of the Institute for Intelligent Information and Data Mining. His research concerns the data mining and machine learning methods in specific backgrounds especially for the astronomical big data. He is a member of China Computer Federation (CCF) and Chinese Astronomical Society (CAS).

XIUJUN ZHAO received the M.S. degree in computer science and technology from the Taiyuan University of Science and Technology, China, where he is currently pursuing the Ph.D. degree. His research interests include data mining and parallel computing.  

...