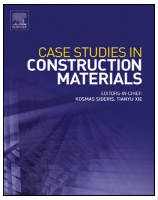
Case Studies in Construction Materials 23 (2025) e04987

Contents lists available at ScienceDirect

Case Studies in Construction Materials

journal homepage: www.elsevier.com/locate/cscm



Application study on conditional generative adversarial network (cGAN) to generate ballast particles for discrete element method simulation

Viet Dinh Le, Gyu-Hyun Go [*]

School of Architecture, Civil and Environmental Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk 39177, Republic of Korea

ARTICLE INFO

Keywords:
Ballast particle
Generative adversarial networks
Particle generation
Generator
Discriminator

ABSTRACT

Understanding ballast particle morphology is important for evaluating the load-bearing capacity of ballasted track foundations using numerical simulations. Although several methods, such as digital imaging, laser scanning, and computed tomography scans, are widely used to capture ballast morphology, real ballast layers consist of tens of thousands to millions of particles of varying shapes and sizes, making these methods complex. Therefore, an efficient solution needs to be found that can generate large-scale ballast datasets for the simulations. This study aims to develop a conditional generative adversarial network (cGAN) to generate ballast particles classified as: angular, subangular, subrounded, or rounded. The cGAN model consists of a generator and a discriminator network, where the generator aims to produce generative data based on the distinction from real data estimated by the discriminator. To find the optimal network architecture for the cGAN, the energy distance metric was investigated by varying the learning rate and number of neurons in hidden layers. The ballast particles generated using the optimal cGAN model were assessed using the receiver operating characteristic area under the curve (ROC AUC). The average ROC AUC was 0.9827, indicating a high classification performance. In addition, roundness coefficients were computed, showing that the morphology of ballast particles generated aligned well with the predefined ballast classes. The cGAN model was therefore shown to be effective at creating realistic ballast particles for further numerical simulations.

1. Introduction

Ballasted tracks are one of the most common structures in railway systems owing to their many outstanding advantages. They provide high stability for superstructures, such as sleepers and rails. They allow an even distribution of the train load to the ground, thereby minimizing vibrations and improving safety when the train is moving at a high speed. In addition, most railways have ballast particles, which help drain water effectively, avoiding water stagnation under the subgrade and thereby reducing the risk of damage due to bad weather, such as flooding or freezing. Another advantage is their ability to reduce vibrations and noise. When the train moves, the ballast layer helps absorb part of the vibration, thereby reducing the oscillation of the train car. Thus, passengers feel more comfortable when the train is moving at a high speed. Furthermore, the maintenance and repair of railways using ballasted particles are simpler. Ballast can be easily added or replaced without requiring expensive labor, allowing maintenance tasks to quickly access

[*] Corresponding author.
E-mail address: gyuhyungo@kumoh.ac.kr (G.-H. Go).

https://doi.org/10.1016/j.cscm.2025.e04987
Received 20 January 2025; Received in revised form 27 May 2025; Accepted 27 June 2025
Available online 4 July 2025

and fix problems. The low cost and high availability of ballast are other important reasons for the wide adoption of ballast tracks for railway construction.

Ballast particles play an important role in the performance of ballasted track systems. The morphological features (shape and size) of ballast particles have a considerable influence the mechanical properties of ballasted track systems. Numerous studies have investigated the shape and size of ballast affect performance under repeated loading by railway structures [7,9,28,37]. The findings indicate that the shape and size of the ballast not only affect the load distribution capacity but also determine the durability and service life of the track foundation under loading cycles. These studies have determined the importance of optimizing the size and shape of ballast particles to enhance the efficiency and durability of railway foundations.

When analyzing the morphological features of ballast particles, it is important to consider their surface texture and angularity, as these factors significantly affect the bearing capacity of the track foundation. Numerical methods, such as the discrete element method (DEM), are commonly used for simulating these factors. To ensure an accurate simulation, the geometric properties of the ballast particles must be digitized and input into the simulation software to create particles during the model initialization process [6,18]. In the DEM model, the calibration of material and geometric properties of ballast particles is essential to improve the accuracy of the ballasted track simulation model. Guo et al. [9] found that ballast degradation is directly related to ballast particle morphology, specifically, the flaky or elongated ballast particles lose volume and sharp corners more readily than cubic ballast particles. They investigated the ballast morphology using the Abrasive Depth method based on Los Angeles Abrasive (LAA) tests combined with three-dimensional (3D) image analysis. Furthermore, key metrics such as Mean abrasive depth and 3D true sphericity are adapted to analyze the ballast particle behavior. He and his colleagues suggested using 3D image analysis in DEM in future work. In addition, Aela et al. [1] investigated the ballast shoulder width and track super-elevation using the DEM model. As a result, they demonstrated to effect of ballasted track geometry on improving the sleeper transverse resistance with wider ballast shoulders (300–500 mm) and increased super-elevation (50–150 mm). These findings highlight the critical factor between particle morphology and track design in improving the accuracy of the ballasted track model. Moreover, Chen et al. [4] used DEM to investigate the effect of particle angularity on the deformation and degradation of ballasted tracks under repeated loading. The ballast particles were analyzed using computed tomography (CT) scanning technology, in combination with image processing, allowing for the reconstruction of the abnormal grains within the DEM. The findings indicate that the grain degradation process was incorporated to simulate particle angularity, thereby enhancing the realism of the ballasted track model.

Several techniques are commonly used to determine the morphological properties of ballast particles. These include traditional photography [9], 3D laser scanning [13,15,38], and X-ray CT technology [4]. Traditional photography involves capturing two-dimensional (2D) images of ballast particles from different angles using digital cameras. The images are then analyzed using image processing software to generate the contours of the particles for the simulation model. Furthermore, 3D laser scanning is utilized to generate a 3D geometric database of ballast particles, while X-ray CT technology allows for the precise identification of geometric properties with high accuracy. All three methods provide the necessary geometric data for the granular particles in the DEM model, which form the ballast layer of the track foundation simulation model.

The combination of scanning techniques for ballast particles and the DEM numerical simulation mode has been successful in analyzing and understanding the working state of the ballast layer in railway foundations. Nonetheless, a ballast layer is composed of numerous particles that vary in type, shape, and size, potentially reaching into the thousands or millions. Creating realistic particles within a DEM model is challenging. Therefore, it is important to enhance the morphological characteristics of ballast particles for the random generation of particle shapes and the extraction of features. In addition, the irregular and complex nature of the actual contours of ballast particles requires simplification to improve the computational efficiency of DEM numerical simulation models. Conventional methods of morphological analysis that use digital image processing techniques frequently depend on fundamental mathematical assumptions, leading to morphological indices that might not effectively represent the actual contours of ballast particles.

The irregular and random nature of ballast particle contours renders accurate scanning and reproduction complex and inefficient. Although advanced techniques such as laser scanning, CT scanning, and sophisticated image processing have been used to accurately capture the irregularity contours of ballast particles, they still encounter challenges in balancing the diversity of grain shapes and simulation efficiency, particularly when using a DEM. To overcome these issues, several methods based on theoretical approaches have been used to reconstruct the ballast particle contours. For example, Tahmasebi [33] combined a level-set algorithm and Markov characteristics to generate complex-shaped particles. Wettimuny and Penumadu [36] used Fourier transforms for digital images to analyzing the shapes of fine and coarse aggregate particles. Mollon and Zhao [23] integrated random techniques with a Fourier shape description to simulate the grain shape and construct a framework for reconstructing particles with complicated shape. Zhou et al. [42] proposed a spherical harmonic analysis to reconstruct contours of sand particle and estimate the shape properties such as roundness coefficient, sphericity, and particle size. Liu et al. [17] proposed a method to characterize the 2D projection of ballast particles based on surface curves and optimal ellipses, and then reconstructed this projection using ballast particle reconstruction function libraries.

Previous studies have successful generated contours of ballast particle using mathematical and statistical approaches. With the advance in machine learning, neural network models have shown great promise in recognizing features and in their ability to learn from images. Some studies have used machine learning to analyze the grain morphology of construction materials. For example, Kim et al. [14] determined the shape parameters of sand particles using deep learning approach. Kim and Youn [12] used conventional neural network models to analyze the roundness and sphericity coefficients of sand and analyzed the data imbalance. In addition, generative adversarial networks (GANs) offer a robust representation learning method that requires minimal labeled data and operates based on the competition between two networks to generate ballast particles. This is demonstrated by the BallastGAN model that was developed by Wang et al. [35] to generate ballast grain contours by using digital images. Consequently, in this study, GANs are adapted to generate ballast particles.

In this study, data described by Chen et al. [4] were used with angular, subangular, subrounded, and rounded ballast classes. The ballast particle images were converted into data vectors, which provided input to the GANs model. This differs from the study by Wang et al. [35], in which images were used as input data. An additional condition was added to the GANs model to ensure that the generated ballast particles fit into one of the four morphological particle classes, giving rise to the conditional GAN (cGAN) that was used in this study.

One challenge is to determine the optimal network architecture for the generator and discriminator networks in a cGAN. This study investigated the influence of the learning rate and the number of neurons in the hidden layer to determine the optimal architecture of the neuron networks. The additional information required by the model is the ballast class label, which allows the cGAN to generate ballast particles with the same morphological characteristics. This has not been investigated in previous studies. Afterward, the performance of the cGAN model was evaluated using two methods: plotting the receiver operating characteristic curve (ROC) and then determining the area under the curve (AUC); and classification based on the roundness coefficient from the ballast particle data generated by the cGAN. The ROC AUC method, which is popular for evaluating classification models, helps to evaluate whether the model generates data that are consistent with the input conditions. Meanwhile, calculating the roundness coefficient helps to confirm that the distribution of the roundness values of the generated ballast class is concentrated in conventional value domains. An automatic roundness coefficient calculation algorithm was developed with several steps, such as noise filtering, key-point identification, corner and non-corner point classification, inscribed tangent circle identification, and roundness coefficient calculation.

The remainder of this paper is organized as follows. Section 2 describes the fundamentals of GANs and cGAN. It also discusses the assessment measures for the cGAN. In Section 3, the development of a cGAN for generating ballast particles and determining the optimal neural network architecture for the generator and discriminator are described. Section 4 presents the performance of the cGAN using the ROC AUC method and calculates the roundness coefficient of the ballast particles generated. The key findings are summarized and the conclusions are presented in Section 5.

2. GANs

2.1. Fundamental of GANs

GAN models have many applications in artificial intelligence (AI), particularly in data generation. Goodfellow et al. [8] first proposed and defined a GAN model. In general, GANs consist of a generator and discriminator networks. The generator creates new data samples from random noise, while the discriminator calculates the probability of the output value to distinguish between “real” and “fake” data generated by the generator. These two models are trained simultaneously in an “adversarial game,” where the generator aims to "fool" the discriminator by producing data samples increasingly like real data, while the discriminator strives to become better at distinguishing between “real” and “fake” data. Recently, the GAN model has been widely applied in many AI applications including image, video, and music generation.

The generator and discriminator are built using deep neural networks. The data is generated by the generator from an input vector, which is a vector of random probability distributions, such as Gaussian or uniform distributions, while the discriminator distinguishes between data generated by the generator and real data from datasets. The discriminator calculates the probability of determining whether the data generated by the generator is “real” or “fake” with probabilities ranging from 0–1. When the probability calculated by the discriminator is close to one, the data generated by the GAN are more like the real data. Consequently, the weight and bias matrices of the generator model are updated throughout the training epoch using the predicted output value from the discriminator model.

During the training process, the generator is trained to create data that confuses the discriminator between “real” and “fake” data. Therefore, the generator aims to minimize its objective function as much as possible according to:

$$\min_{G} V_G(D,G) = E_{z\sim P_z}[\log(1 - D(G(z)))] \tag{1}$$

In contrast, the discriminator is distinguished between the data generated by the generator and the real data. Therefore, the discriminator aims to maximize its objective function, and is represented as:

$$\max_{G} V_G(D,G) = E_{x\sim P_t}[\log D(x) + E_{z\sim P_z}[\log(1 - D(G(z)))]] \tag{2}$$

where $D(x)$ is the predicted probability of discriminator detecting the real data vector x; $G(z)$ is the generated data created by the generator from the noise vector z; and P_t and P_z are the distribution of real data x and noise vector z, respectively.

Finally, the objective function of the GAN model is a combination of the objective functions of the generator and the discriminator, and is expressed as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

The generator and discriminator collaborate during the training process to generate data that are close to real data. Although a GAN can generate new data, model optimization is essential when using data containing multiple distinct groups. Among the various GAN variants, the cGAN model is an ideal solution for generating particles based on the specific angular, subangular, subrounded, and rounded categories. Therefore, the cGAN model was adapted to generate ballast particle based on specific particle-shape conditions.

Table 1
Comparison of cGAN and traditional methods for generating ballast particle data.

Method	Advantages	Disadvantages
cGAN	Rapidly generates ballast particle shapes for DEM with a large number of samples. Flexible and diverse processing due to input particle class conditions. Highly scalable for large datasets.	Difficult to balance between generator and discriminator networks. May not generate particle shape probability accurately.
GANs	Quickly generates ballast morphology similar to input data. Simple training process for large datasets. Streamlined sampling process.	Cannot control specific particle morphology due to lack of conditional inputs. Difficult to adjust proportions of different particle shapes in the model.
CT Scanning	Very accurate in capturing realistic ballast shapes. Provides detailed 3D data for small samples, ensuring accuracy of particle grains in DEM models.	Slow and costly generate large datasets. Requires specialized equipment and extensive processing.
3D imaging	Accurate in capturing detailed ballast surface textures. Suitable for high-quality, small-scale datasets and models.	Time-consuming and expensive for large datasets. Requires complex image processing and is not scalable for large datasets

2.2. cGAN

The cGAN model, developed by Mirza and Osindero [22], is an extension of the GAN model. Conditional information was added to both the generator and discriminator [22]. This conditional information can be in the form of class labels, attributes, or other data that can constrain the cGAN to generate data based on specific conditions. Consequently, a cGAN can generate conditional synthetic data, meaning that the characteristics of the output data can be controlled based on specific conditional input information. This technique enhances the practical applications of the cGAN model for generating structured and group-specific data.

To integrate the conditional information into the cGAN, an input vector was created for both the generator and discriminator models. The input data for the generator comprises a random noise vector z, and conditional information y. Similarly, the input data for the discriminator includes both real and fake data combined with the conditional information y. The objective function of the generator is therefore rewritten to incorporate the conditional information y as:

$$\min_G V_G(D,G) = E_{z\sim P_z(z),y\sim P_{data}(y)}\left[\log(1-D(G(z,y),y))\right] \tag{4}$$

To avoid the above objective function becoming a vanishing problem, it is often adjusted to:

$$\min_G V_G(D,G) = E_{z\sim P_z(z),y\sim P_{data}(y)}\left[-\log D(G(z,y),y)\right] \tag{5}$$

Similarly, the objective function of the discriminator is rewritten to include the vector y as follows:

$$\max_G V_G(D,G) = E_{(x,y)\sim P_{data}(x,y)}\left[\log D(x,y) + E_{z\sim P_z,y\sim P_{data}(y)}\left[\log(1-D(G(z,y),y))\right]\right] \tag{6}$$

In order to generate particles for the DEM model, cGAN is selected due to its advantages in generating a large and diverse set of ballast particles. The comparison of cGAN and traditional methods is shown in Table 1. Compared to other methods, cGAN can quickly generate thousands or millions of ballast particles with accurate shape classifications (angular, subangular, subrounded, rounded). The capability to manipulate particle shapes based on class information conditions provides cGAN with the flexibility to generate diverse shape needs, allowing for the requirements of large-scale DEM simulations. In contrast, traditional GAN models can generate ballast particles quickly, but they cannot control specific particle shapes, making it difficult to adjust the proportion of particle types, for example, the ratio of rounded over angular ballast particles, which reduces simulation accuracy. Meanwhile, CT scanning and 3D imaging are provided highly accurate 3D data and surface textures for small ballast samples; however, these techniques are slow, costly, and require complex processing, making them unsuitable for large datasets with thousands or millions of particles needed for DEM. Although cGAN has challenges in balancing the generator and discriminator leading to inconsistent results. In addition, this study proposed a procedure to find the optimal network architecture, adjusting parameters such as learning rate and the number of neurons in the generator and discriminator in cGAN. Additionally, cGAN may not generate ballast particles with an exact roundness coefficient according to standard particle types. However, the main scope of this study is to create near-realistic particle samples to DEM simulation models, rather than achieving precise probability distributions for each particle type. Furthermore, the generated ballast particles can be controlled and filtered based on roundness coefficients resulting in a diversity of particles in the DEM model.

In this study, the ballast particle data were divided into angular, subangular, subrounded, and rounded groups to simulate different particle shapes. Each of these groups represents a wide range of morphological and structural characteristics of natural ballast particles. To accurately simulate the ballast particles within each category, the cGAN model was adapted to generate specific particles based on conditional information. In this approach, the index number of the ballast particle group was used as a conditional input for the cGAN. This approach allows the model to optimize the generated particles to precisely match the morphological and structural characteristics of each specific particle group.

2.3. Evaluation technique for cGAN

In general, a cGAN aims to train a specialized generator capable of producing synthetic samples that closely match the distribution of real samples. Consequently, it is important to evaluate the generator performance. The evaluation metrics for cGAN models can be used for early stopping and to determine the optimal neural network architecture for both the generator and discriminator. Most of the current cGAN evaluation metrics, such as the maximum mean discrepancy [5,26], 1-nearest neighbor classifier [19], least squares loss [20], Wasserstein distance [2,39], and energy distance [29,40], are based on the loss function between real and generated data by the cGAN model. Among these, energy distance is a measure with desirable properties, such as being continuously differentiable and satisfying the four properties of probability distance, which are symmetry, relaxed triangle inequality, identity of indiscernibility, and non-negativity. In addition, it assists in evaluating the quality of the generated data in each training epoch to ascertain whether they meet the acceptance standard in the testing procedure using the distribution of real values.

The concept of "energy distance" was first introduced by Székely [29] while teaching at several institutions including Budapest, and MIT, Yale, and Columbia in US. The main concept stems from Newton's potential energy and has become widely used for statistical observations in metric spaces. Energy statistics are functions that depend on the distance between statistical observations and are only zero when the statistical hypothesis is true. In addition, the energy distance has been studied using machine learning theory [27]. As a result, the "energy distance" between two probability distributions P_{data} and P_z [16,30–32] can be expressed as:

$$ED(P_{data},P_z) = 2\boldsymbol{E}_{x\sim P_{data},z\sim P_z}\|x-z\| - \boldsymbol{E}_{x\sim P_{data},x'\sim P_{data}}\|x-x'\| - \boldsymbol{E}_{z\sim P_z,z'\sim P_z}\|z-z'\| \tag{7}$$

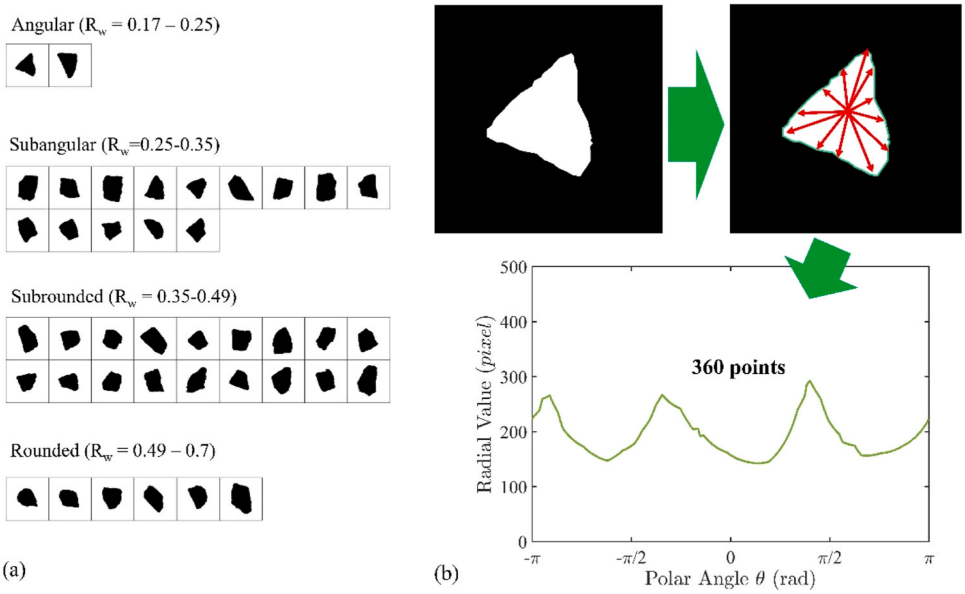

Angular (R_w = 0.17 – 0.25)
Subangular (R_w=0.25-0.35)
Subrounded (R_w = 0.35-0.49)
Rounded (R_w = 0.49 – 0.7)
(a)
(b)
360 points
Radial Value (pixel)
Polar Angle θ (rad)
500
400
300
200
100
0
-π
-π/2
0
π/2
π

Fig. 1. Database preparation: (a) Ballast particles [4]; (b) Particle irregularity generation based on image processing.

In this study, the shape of a ballast particle was represented as a data vector instead of using images, which is the most common data form used in other studies. As a result, the "energy distance" metric is suggested to evaluate the performance of the cGAN model at finding the neural network architecture of both the generator and discriminator. In addition, an early stopping rule was adopted to avoid overfitting and underfitting during the cGAN training process. Upon completion of the training process, the trained model is typically evaluated using several metrics. Among these, the ROC AUC is commonly used to evaluate the performance of the classification and data generation models. Therefore, in this study the ROC AUC was adopted to evaluate the performance of the cGAN models, which generated ballast particles using input noise vectors and specified ballast group labels (angular, subangular, subrounded, and rounded).

The ROC AUC enabled the evaluation of the model's ability to distinguish between different ballast particle groups. The ROC value was determined by measuring the true-positive rate (TPR) and false-positive rate (FPR) at different thresholds. Additionally, the AUC provides a comprehensive measure of cGAN accuracy. A value close to one signifies a strong classification capability between ballast particle groups, whereas a value close to 0.5 indicates poor performance and almost random selection. Furthermore, the ROC AUC can determine misclassified ballast particles, aiding in error classification and shape correction during model training. This ensures that the generated ballast particles align accurately with the specific angular, subangular, subrounded, and rounded groups.

By contrast, the ROC AUC provides detailed insights into the sensitivity and specificity of the cGAN model. This enables the performance of the cGAN model in different scenarios to be understood, specifically when a certain ballast class is underrepresented compared to others. This approach is useful for improving and optimizing the model by focusing on minimizing the misclassification errors in a specific ballast group. Therefore, the ROC AUC is a powerful method for evaluating and enhancing the cGAN model for generating ballast particles. This approach ensures that the model not only generates high-quality particles but also accurately meets the classification requirements.

3. Development of cGANs for generating ballast particles

3.1. Data preparation

In this study, the ballast particle datasets were extracted from a study by Chen et al. [4]. Ballast particles were classified into angular, subangular, subrounded, and rounded groups based on their roundness coefficients, R_w, as shown in Fig. 1(a). Angular particles are characterized by sharp edges and irregular shapes. The subangular group exhibits fewer sharp edges and smoother shapes. Subrounded particles have more rounded edges, with a shape that is nearly round, but still retain some minor angular features. Finally, the rounded particles exhibit an approximately perfect round shape with smooth edges and no distinct angular features.

Based on the work of Chen et al. [4], the ballast samples were scanned using a CT machine and converted into high-quality images. Forty ballast samples were examined, comprising 2 angular, 14 subangular, 18 subrounded, and 6 rounded samples. This study aims to develop a model that can generate more ballast samples in the quantities specified by the user. To ensure that the training data covered a wide range of shape and size characteristics, the four ballast particle groups from Chen's study were redrawn to create the datasets for this study.

In this study, the OpenCV library, which is a robust open-source library for image processing analysis, was used to create the dataset

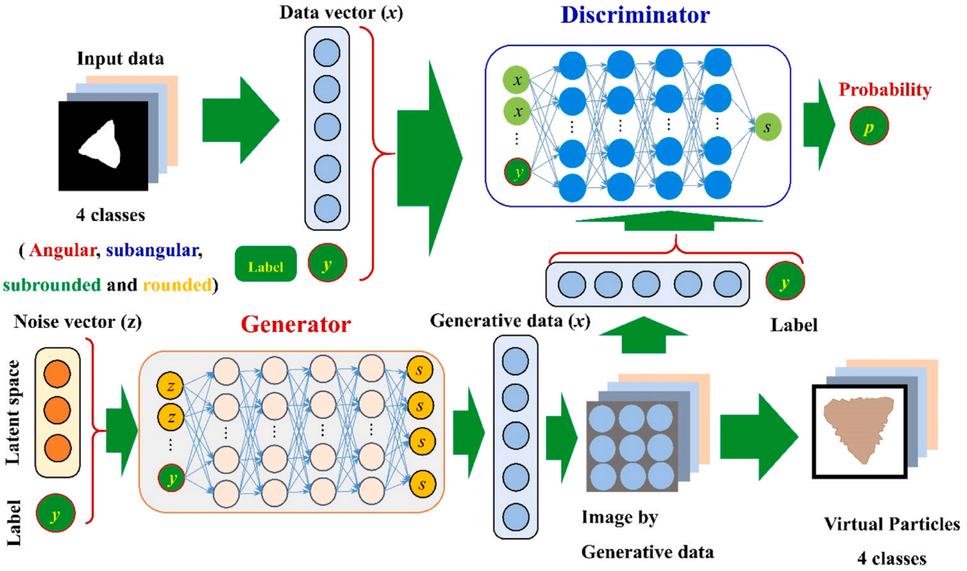


Fig. 2. Schematic of cGAN for particle generation.

for the cGAN model. Generally, an image JPG file has a common format and can be read using OpenCV functions. The images were then transformed to grayscale using the *cv2.cvtColor()* function, thereby reducing unnecessary information and focusing on the intensity of the pixels as shown in Fig. 1(b). In the original image, a pixel is typically defined by R, G, and B values, corresponding to the red, green, and blue primary colors, respectively. Following the International Telecommunication Union standard, the grayscale value I was calculated for each pixel using:

$$I = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{8}$$

where R, *G* and B are values in the range from 0 to 255, representing the three primary colors red, green, and blue, respectively.

Otsu's method, which is an automatic thresholding technique, was then applied to identify the optimal threshold for distinguishing between an image background and an object. This was achieved using the *cv2.threshold()* function in OpenCV, which was specifically designed based on Otsu's method to maximize the objective function of the pixel distribution as follows:

$$\sigma_B^2(T) = w_1(T) \times w_2(T) \times [\mu_1(T) - \mu_2(T)]^2 \tag{9}$$

where: *T* is the threshold value, an integer ranging from 0 to 255. The weights $w_1(T)$ and $w_2(T)$ are the probabilities of the two classes separated by the threshold *T*; the parameters $\mu_1(T)$ and $\mu_2(T)$ are the means of the pixel values for the two classes; and the threshold *T* is a predetermined value used to convert a grayscale image into a binary image, where the pixels belonging to the object are categorized as white and those belonging to the background are categorized as black. Following the conversion of the image to binary, the values of the binary matrix are inverted to meet the analysis requirements using the *cv2.bitwise_not()* function from OpenCV.

To extract the geometric features of the particle in the binary image, the *cv2.findContours()* function from OpenCV is applied to identify the irregular polyline of the particle, allowing the coordinates of the boundary points and the shape of the ballast particle to be obtained. In addition, the *cv2.moments()* function is used to calculate the geometric center of the particle. The high-resolution image of ballast particles from the study by Chen et al. [4] contained many boundary data points, typically ranging from several hundred to thousand. To reduce the computational cost of the data preparation, the original data points were down sampled to 360 points, corresponding to 360° in the angular coordinate system, using an interpolation method. Consequently, the input data to the cGAN model are numeric vectors representing the arbitrary radii of the points on the boundary of the particle, corresponding to angular values from 0 to 360° as shown in Fig. 1(b).

3.2. Building the cGAN model

This study aims to develop a cGAN to generate ballast particles with the index number of particle class as an input condition. Fig. 2 illustrates the structure of the cGAN model, which comprises a generator and discriminator networks. The generator creates synthetic data for ballast particles based on the input condition, whereas the discriminator is responsible for distinguishing between synthetic and real data from the datasets. Throughout the training process, the generator aims to generate new data that closely resemble real data from datasets based on feedback from the discriminator.

The generator network receives the noise vector and label vector as input data. The label vector represents a specific ballast particle class and is expressed as a data label. Additionally, the label vector is represented in a binary format consisting of four bits. For

Table 2
Several approaches for determining the number of neurons in the hidden layer.

ID	Number of neurons*	Citation
1	$2N_i + 1$	[3]
2	$\frac{N_o + N_i}{2}$	[25]
3	$\sqrt{N_i \times N_o}$	[21]
4	$\frac{2 + N_o \times N_i + 0.5N_o \times (N_o^2 + N_i) - 3}{N_o + N_i}$	[24]
5	$\frac{2N_i}{3}$	[34]
6	$2N_i$	[11]
7	$2\sqrt{(N_o + 2)N_s}$	[10]
8	$\sqrt{(N_o + 2)N_s} + 2\sqrt{\frac{N_s}{(N_o + 2)}}$	[10]

[*] N_s is the training sample; N_i and N_o are the number of neurons in the input and output layer, respectively.

Table 3
Proposal of number neurons in the hidden layer for generator and discriminator networks.

Case	Number neurons in the hidden layer			
	Hidden layer 01	Hidden layer 02	Hidden layer 03	Hidden layer 04
1	128	256	512	1024
2	128	256	1024	512
3	128	512	256	1024
4	128	1024	512	256
5	256	512	1024	128
6	256	1024	512	128
7	512	1024	256	128
8	1024	512	256	128

example, label vectors for the angular, subangular, subrounded, and rounded classes are defined as vectors [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], and [0, 0, 0, 1], respectively. The noise vector was randomly generated based on a Gaussian distribution with values ranging from 0 to 1. The noise vector was selected to have 100 elements, whereas the label vector contained four elements. Therefore, 104 neurons were present in the input layer of the generator network. The output of the generator network was a vector containing 360 elements corresponding to an arbitrary radius of the ballast particle, with angular coordinates ranging from 0 to 360°.

On the other hand, the discriminator network has two training steps using both real data from the dataset and data generated by the generator network. It considers a vector of 360 arbitrary radius values of the ballast particle combined with a label vector representing the ballast particle class. However, the label vector for the discriminator is the actual label from the dataset. Consequently, the input layer of the discriminator network consists of 364 neurons, and the output layer includes a single neuron that represents the probability of distinguishing between real and generative data. The probability values range from 0 to 1. A probability value close to zero indicates that the generated data differs significantly from the real data, whereas a probability value close to one suggests that the generated data closely resembles the real data, making it difficult for the discriminator to distinguish between the two sets of data.

In general, both generator and discriminator networks typically consist of the input layer, hidden layer(s), and the output layer. The performance of both models was influenced by the number of neurons in the hidden layer(s) and the number of hidden layers. It is important to determine the optimal neural network architecture for the generator and discriminator. The optimal neural network architecture enhances the ability of the generator to generate accurate particles that satisfy the specified conditions of the ballast class and improves the capacity of the discriminator to classify real and generated data.

Many previous studies have recommended approaches for determining the number of neurons in the hidden layers. Table 2 lists the different calculation formulas from the cited references shown. However, we propose a new approach for selecting the number of neurons based on exponential values, such as 2^3 and 2^4, while using the remaining four hidden layers for both the generator and discriminator networks.

Table 3 lists the proposed options for the number of neurons in the hidden layers of the generator and discriminator networks. Moreover, the learning rate significantly affects the performance of the neural network model. A high learning rate may result in underfitting, whereas a lower learning rate can enhance accuracy. We propose a range of learning rates:10^{-5}, 10^{-4}, 10^{-3}, 10^{-2} and 10^{-1}. For each learning rate, we considered 64 cGAN models based on different combinations of the number of neurons in the hidden layers for both the generator and discriminator networks, resulting in 320 models. In addition, we used an early stopping technique to stop the training epoch early for each model and evaluated the performance using the energy distance metric. The optimal configuration was determined using the smallest energy–distance value. The energy distance values for the cGAN models after training the 320

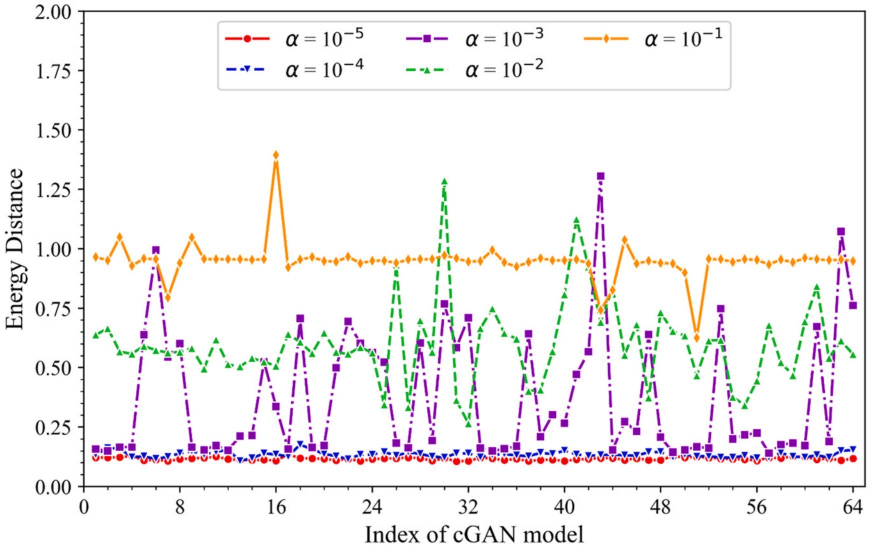


Fig. 3. Evolution of energy distance with different architecture of cGAN.

Table 4
Architecture networks for generator and discriminator of optimal cGAN model.

Network	Number neurons in the input layer	Number neurons in the hidden layer				Number neurons in the input layer
		Layer 01	Layer 02	Layer 03	Layer 04	
Generator	104	128	1024	512	256	364
Discriminator	364	512	1024	256	128	1

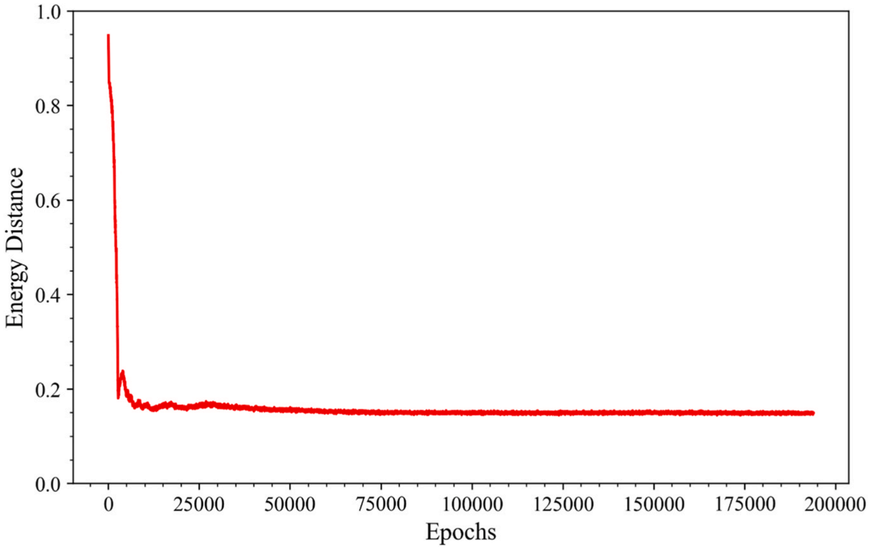


Fig. 4. Energy distance and epoch of selected cGAN model.

models are shown in Fig. 3. Based on the energy distance values among the models, we identified the optimal cGAN model to be the 32nd with the learning rate of 10^{-5}. The network architecture shown in Table 4.

Fig. 4 shows the change in energy distance throughout the training epochs of the optimal cGAN. Initially, the energy distance was relatively high and decreased significantly within the first 500 epochs. It then fluctuated over the next 1000 epochs, gradually decreasing and stabilizing from the 25,000th epoch. To maintain model stability, a patience value of 10,000 steps was set to prevent overfitting or underfitting, resulting in the model being stopped at the 195,000th epoch. Consequently, the optimal cGAN model not

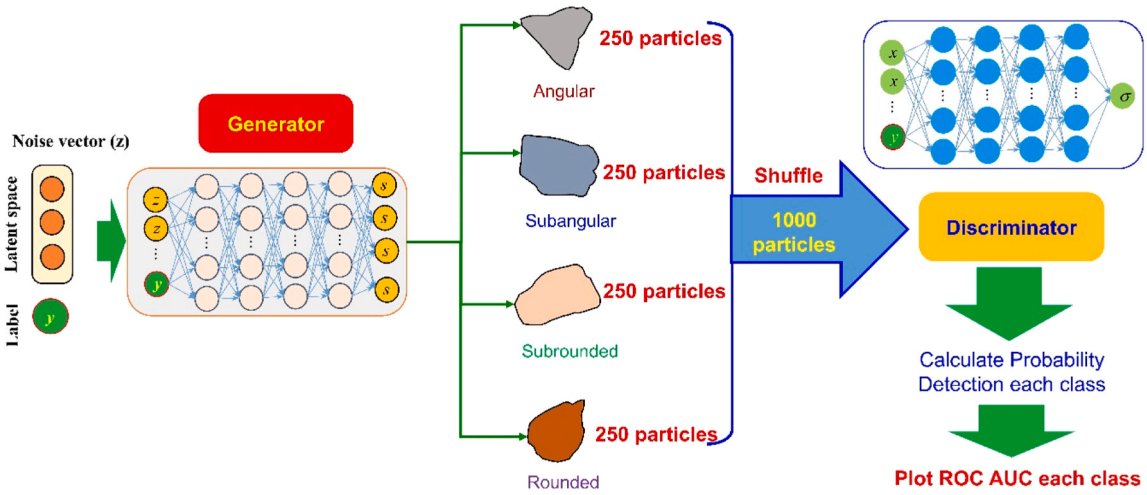


Fig. 5. Schematic of particle generation for the ROC AUC method.

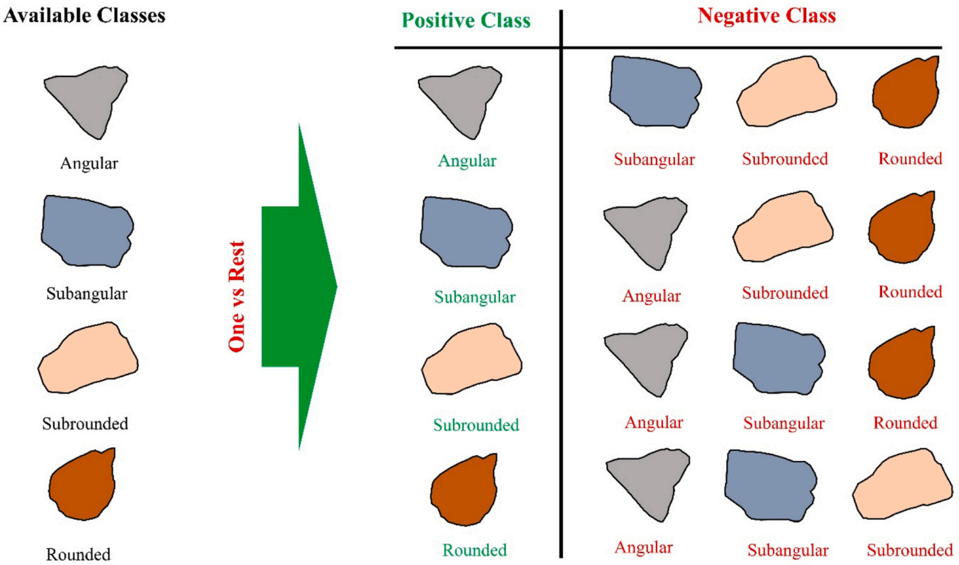


Fig. 6. Combination class following OvR method.

only meets the specific requirements for the ballast particle class, but also shows high efficiency in generating ballast particles under the given conditions. The optimal cGAN model demonstrated the capability of generating high-quality and accurate ballast particles, indicating its potential for practical applications in ballast particle generation. Furthermore, the pretrained model is evaluated using the ROC AUC metric and the roundness coefficient classification method in the following sections.

4. Evaluation of ballast particles generated by cGAN

Two main approaches were used to evaluate the effectiveness of cGAN in creating ballast particles. The first involved using the ROC AUC with the one-vs-rest (OvR) method to assess classification accuracy. The second approach involves determining the roundness coefficients of the generated particles and comparing them with the standard roundness coefficient for each class of ballast particles.

4.1. ROC AUC metric evaluation

To evaluate the performance of the cGAN, we applied the ROC AUC method with the OvR method. The OvR method is a multiclass classification approach that compares each class against all other classes to evaluate the classification capability of the model. Specifically, the ballast particle groups are classified as angular, subangular, subrounded, and rounded. Each class is individually

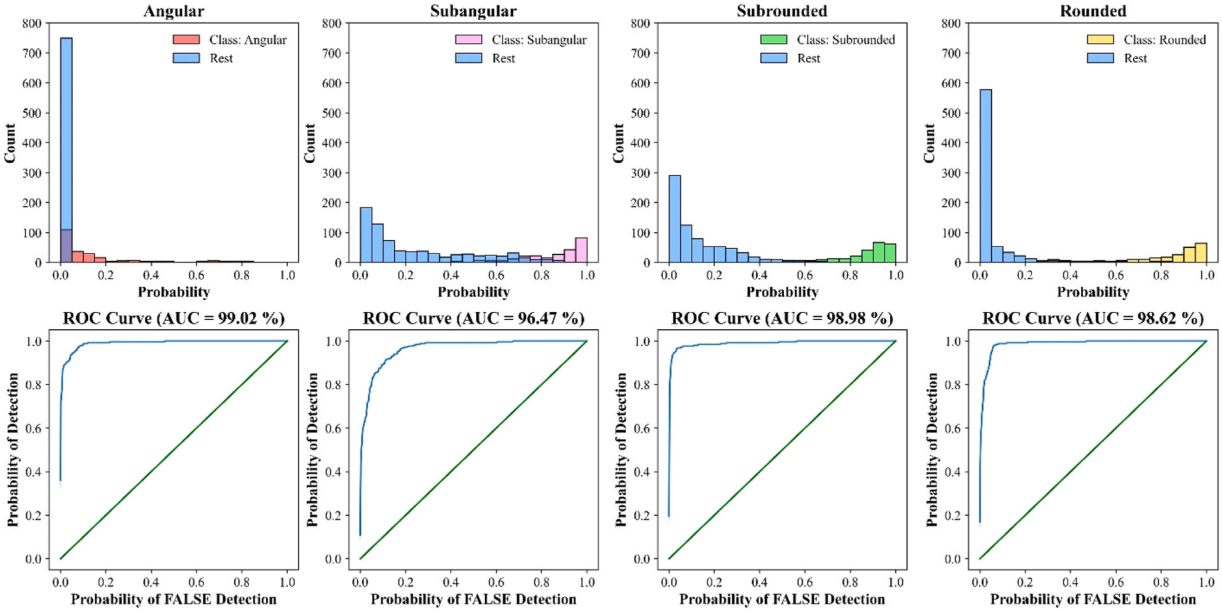


Fig. 7. Histogram diagrams and ROC curves for each particle class.

compared with the other three classes to create separate ROC curves. Fig. 6 shows the application of the OvR method to the four distinct ballast particle classes in this study. The ROC curve and AUC serve as standard metrics for evaluating the effectiveness of the discriminator in the cGAN model in distinguishing between different ballast particle classes. The ROC curve depicts the relationship between the TPR and FPR at different classification thresholds. Meanwhile, the AUC value indicates the capacity of the cGAN model to differentiate between different classes; a value close to one signifies a strong classification performance, whereas a value approaching 0.5 suggests that the model performs close to randomly. This assessment provides a clear understanding of the classification effectiveness of the cGAN model and aids in identifying the strengths and weaknesses of classifying ballast particle classes.

The trained generator weight matrices were used to generate angular, subangular, subrounded and rounded ballast particle classes data for the ROC AUC method. Each class comprised 250 generative particles, resulting in 1000 samples in total as shown in Fig. 5. The samples were then randomized before being input to the discriminator for classification. The weight matrices of the trained discriminator were used to compute the probabilities, which were subsequently used to estimate the ROC AUC metric. The OvR method was implemented to categorize the ballast particle classes as shown in Fig. 6. The discriminator therefore processed 1000 samples applying the OvR method four times, each corresponding to one ballast particle class.

In Fig. 7, the ROC AUC metrics illustrate the classification performance of the cGAN model using the OvR method across the angular, subangular, subrounded, and rounded ballast shapes. In addition, the average of the ROC AUC is 0.9827. These results indicate that the cGAN model demonstrates a high classification performance, particularly with scores of approximately 0.99 for the angular and subrounded groups. Although the subangular class achieved the lowest score of 0.9647, it still demonstrated an effective performance.

The distribution histogram for the angular class indicates that most of the data for this class is associated with low probability values. The cGAN model may struggle to classify samples belonging to this class accurately. One possible explanation for this observation is that the model applied a high classification threshold to ensure accuracy, resulting in numerous angular samples with low probabilities. Although the cGAN model can generate a specified number of samples, only two particles in the datasets were used for the angular class. This limited dataset may have caused the cGAN model to lack sufficient geometric information about the angular particles, thereby making classification more challenging than for the other particle classes.

However, the distribution histograms for the rounded and subrounded classes are clearly differentiated from the others, indicating strong probabilities and effective classification. In addition, the subangular class exhibited a noticeable differentiation, although the probabilities were slightly lower than those of the other classes.

In conclusion, the ROC curves for all classes were close to the y-axis, particularly for the angular and subrounded classes, indicating excellent classification performance. The ROC curve for the subangular class also demonstrated a high classification effectiveness, although it was slightly lower than that of the other classes. These results and analysis show that the cGAN model performs well with high ROC AUC scores for all ballast classes. However, to further enhance the reliability of the model, particularly for the angular class, we recommend the following improvements:

- Ensure a balanced and comprehensive distribution of training data for this class.
- Adjust the classification threshold to improve accuracy.
- Use data augmentation techniques to increase the number of samples in the angular class.

These enhancements will significantly improve the capability of the cGAN, particularly involving particle classes with lower

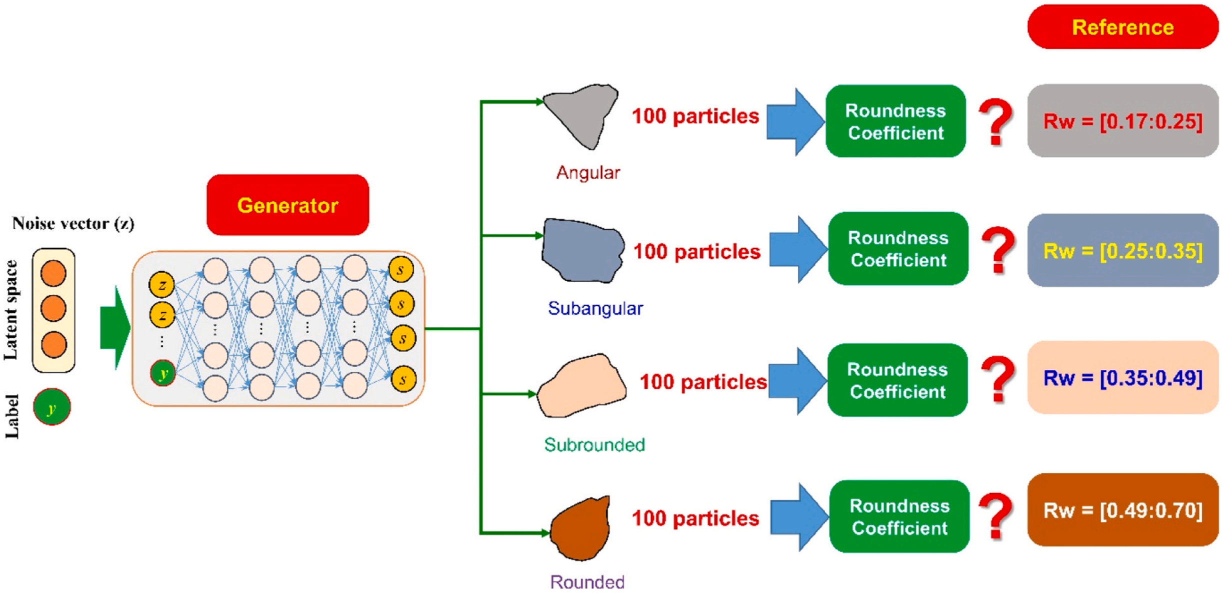


Fig. 8. Schematic of particle generation for roundness coefficient R_w calculation.

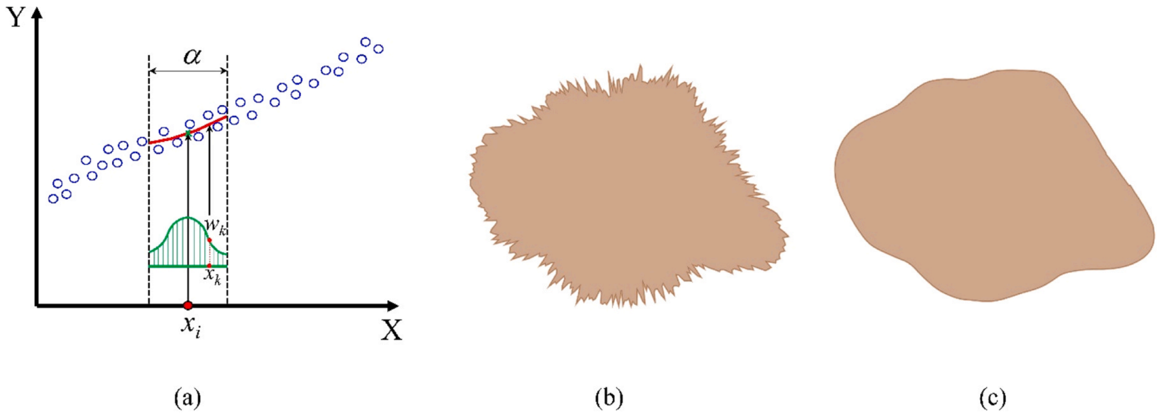


Fig. 9. Post processing of particle generation: (a) LOESS approach; (b) Particle generated by cGAN model; (c) Smoothed particle.

classification probabilities.

4.2. Roundness coefficient of ballast particles calculation

To evaluate the ability of the cGAN model to generate ballast particles with a standard classification, it is important to calculate the roundness coefficient and analyze its distribution. This analysis also provides valuable insights into the diversity of particles generated by the cGAN model. A well-performing model should generate ballast particles with shape characteristics that are evenly distributed across predefined classes, demonstrating the accuracy of the model and its ability to effectively simulate different particle types. This approach enables the capacity of the model to meet practical application requirements. Therefore, potential improvements are suggested to enhance the quality and efficiency of the cGAN model.

To verify the quality of the ballast particles generated using the cGAN model, the roundness coefficient R_w for each particle is computed using:

$$R_w = \frac{\sum_{i=1}^{n} r_i}{n \times R_{insc}} \tag{10}$$

where r_i is the radius of the small inscribed tangent circle, n is the total number of small inscribed tangent circles on the particle. and R_{insc} is the radius of the largest inscribed circle.

Using the trained cGAN model, the weight matrix of the generator is applied to generate data to calculate the roundness coefficient

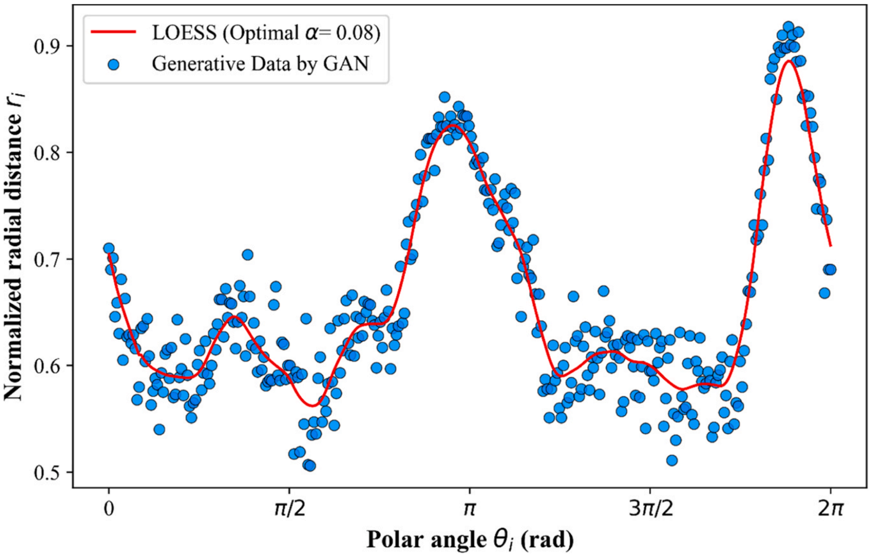


Fig. 10. Comparison of smoothed and generated data using LOESS.

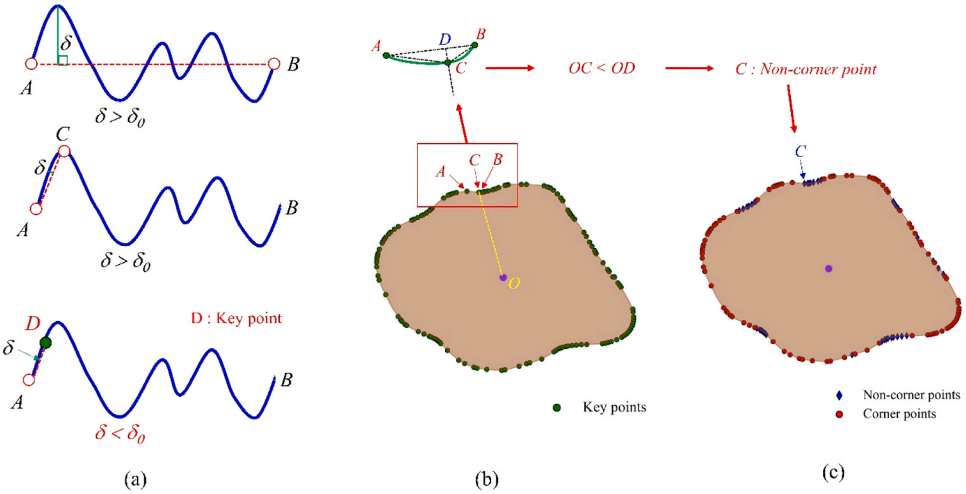


Fig. 11. Corner and non-corner identification: (a) Key points determination; (b) Key points expression; (c) Corner and non-corner points.

of the particles. The calculation process is illustrated in Fig. 8. The generator generates 100 particles for each of the angular, sub-angular, subrounded, and rounded classes. The roundness coefficients of the particles are calculated and compared with the corresponding reference value ranges for each ballast class shown in Fig. 8. Before calculating the roundness coefficient, the noise was removed from the particles generated using the cGAN model. In this study, locally weighted scatterplot smoothing (LOESS) was adopted to ensure smooth particle geometries. Fig. 9(a) illustrates the LOESS algorithm in calculating the weight for point x_i and its neighboring points within a span distance α. For each point within the span α, the weight wk for any given k-th point is calculated using:

$$w_k = \exp\left[-\frac{(x_i - x_k)^2}{2\alpha^2}\right] \tag{11}$$

where α= 0.08 is set. x_i is the coordinate of the group of points within the span α, while x_k is the coordinate of the *k-th* calculated point.

In general, the particles generated by the cGAN model often exhibit significant noise owing to the limitation in the number of data points initially generated by the generator network, which is only 360 points as shown in Fig. 9(b). To increase the number of data points, the particle polygon is interpolated linearly, increasing the number of data points from 360 to 1000. When the LOESS algorithm is applied, the noise is removed, making the ballast particles smoother and more realistic as shown in Fig. 9(c). In addition, Fig. 10 shows a comparison between the smoothed curve-based LOESS and the particle data points generated by the cGAN. The figure shows

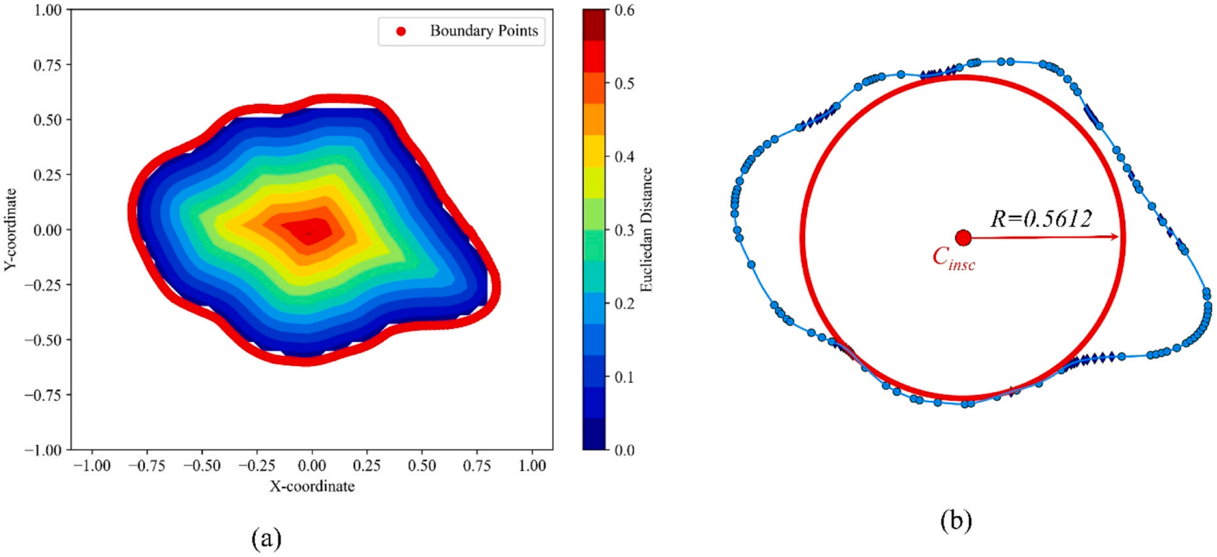


Fig. 12. Determination of inscribed circle: (a) Euclidian Distance map (b) Largest inscribed circle inside particle.

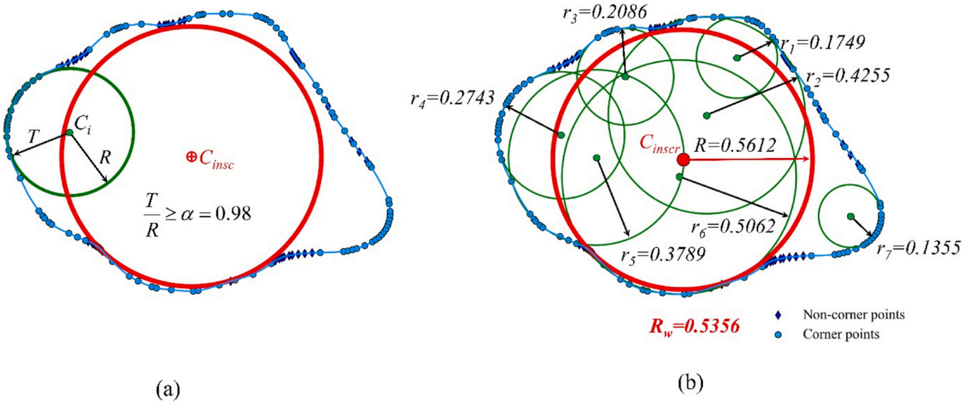


Fig. 13. Roundness calculation procedure: (a) Inscribed tangent circle determination; (b) Inscribed tangent circles and a largest inscribed circle.

that the smoothed curve using LOESS demonstrates the trend of the generative data with acceptable error margins.

Determining the inscribed tangent circles inside the ballast particles is important, and depends on the identification of the key points. Fig. 11(a) illustrates the process of identifying the key points on the boundary curve of a particle, which is divided into segments connected by the key points. The number of key points is determined based on the given condition value δ_0, which is the maximum distance between the two boundary edges. In this study, the δ_0 value was selected as 0.05, which corresponds to 5 % of the maximum arbitrary radius of the particle. Following Fig. 11(b), the key points are estimated and used to classify the corner and non-corner points. Assuming point C and two adjacent points A and B, point D is the intersection of the line connecting the two adjacent points and the extension line passing through point C to the center point of the particle. If OC > OD, then the point C being examined is a corner point; otherwise, point C is not a corner point. Fig. 11(c) shows the non-corner and corner points. After classifying the corner and non-corner points, an algorithm was implemented to determine the inscribed tangent circles at the corner point positions. The detailed calculation process for finding these circles is described by Zheng and Hryciw [41].

By contrast, the largest inscribed circle is an important factor for computing the roundness coefficient of a particle. The center point coordinate of the largest inscribed circle is determined by the coordinates of the pixel that has the maximum Euclidean distance to the nearest boundary point of the particle. The radius of the largest inscribed circle is that maximum Euclidean distance. Fig. 12 illustrates the Euclidean distance map and largest inscribed circle for a particle.

To find the inscribed tangent circles, the minimum distance from the center of the inscribed tangent circles to the boundary of the particle (T) and the desired radius of the circle (R) are determined using the value of α. This value is the maximum acceptable ratio, and $\alpha = 0.98$ was set for this study as shown in Fig. 13(a). The largest inscribed circle and inscribed tangent circles of an example particle are shown in Fig. 13(b). Upon completing this calculation, the roundness coefficients of the particles are estimated using Eq. (10). The roundness coefficients of the particles are calculated and presented corresponding to the angular, subangular, subrounded, and

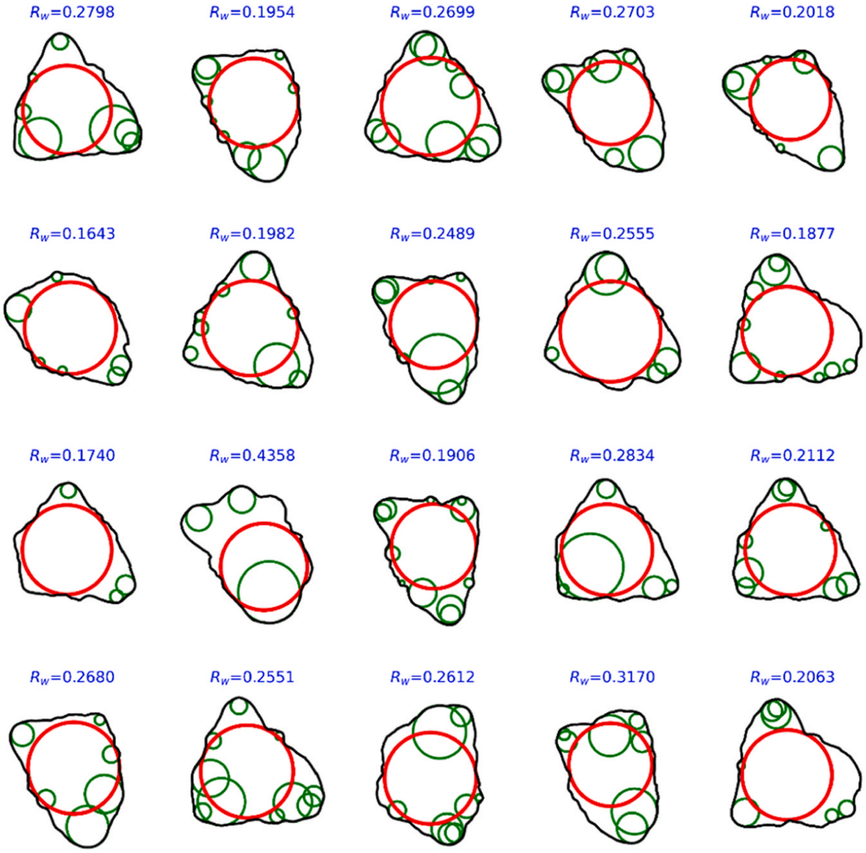


Fig. 14. Roundness calculation for angular particles generated by cGAN.

rounded classes, as shown in Figs. 14–17, respectively.

The roundness coefficient for each ballast particle class was analyzed according to standard value ranges corresponding to the angular, subangular, subrounded, and rounded particle types. This analysis helps to accurately assess the distribution of the roundness coefficients and the ability of the cGAN to generate ballast particles with different shape characteristics. These results not only demonstrate the model's accuracy in generating ballast particles, but also allow for the evaluation of the diversity of the generated particles, thus determining the capability of the model to simulate different types of ballast particles with different geometric properties.

Fig. 18 illustrates the results of the roundness coefficient distribution analysis across the ballast particle classes with the corresponding reference value ranges, according to Chen et al. [4]. The chart consists of a line chart, and a box plot, providing a comprehensive view of how the measured values change across different roundness coefficient (R_w) ranges.

The line chart shows the data series for the angular (blue), subangular (green), subrounded (red), and rounded (purple) particle types. The x-axis is the calculated roundness coefficient (R_w), whereas the y-axis is the measured values in terms of frequency. The shaded background areas in the chart represent the specific reference R_w ranges for each ballast particle class, as suggested by Chen et al. [4]. The corresponding bold lines represent the distribution of the calculated roundness coefficient values for the particles generated by the cGAN model.

According to the line chart, the angular data series had distinct peaks with the highest frequency of 11, and values falling within the conventional range of R_w= [0.17, 0.25]. However, there was an overlap with the subangular class, as the distribution of the roundness coefficient values fell within the reference range of the subangular class, with frequencies ranging from 1–9, and within that of the subrounded class with a frequency < 2. Furthermore, the subangular series fluctuates and has a wide distribution range similar to that of the angular class, but with higher frequencies that mostly lie within the reference range of R_w= [0.25, 0.35]. The calculated roundness coefficients for this group overlap with other particle classes but generally have lower frequencies, ranging from 1–4. Additionally, the subrounded series has a wide distribution range, extending from 0.15–0.6, with overlapping regions in the reference ranges of the angular, subangular, and rounded classes. However, the highest frequency of the calculated roundness coefficients was within the conventional reference range of the subrounded class, R_w= [0.35, 0.49]. Moreover, the rounded series exhibited a more stable pattern, with an even frequency distribution of the calculated values within the conventional reference range of R_w = [0.49, 0.7]. Although there is some overlap, and a few calculated coefficients exceed the reference range, their frequencies are sufficiently low to be acceptable.

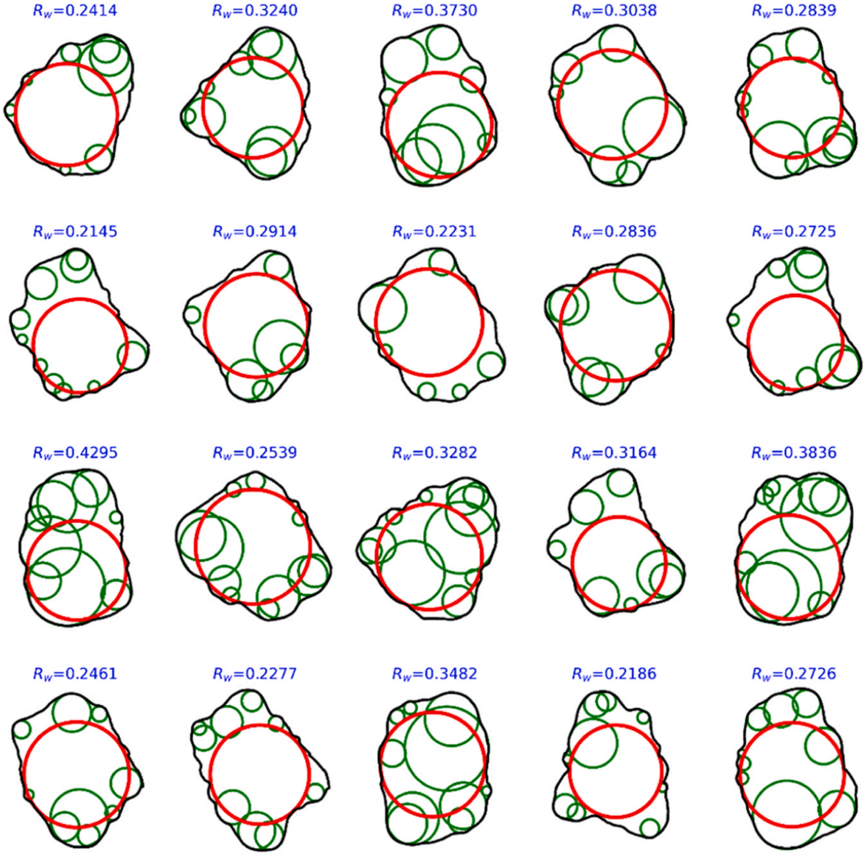


Fig. 15. Roundness calculation for subangular particles generated by cGAN.

The calculated roundness coefficients of the particles generated by the cGAN model showed variability in the data distribution within each class, despite the specific group labels used during data generation. This indicates that the generated data has significant differentiation and diversity in particle shapes within the same class, and that the calculated roundness coefficients do not necessarily conform strictly conform to the conventional values for each particle class.

The box plot in the lower half of Fig. 18 provides an overview of the distribution of the calculated roundness coefficients for the particle classes. This figure provides a more detailed view of the distribution and median values of the measurements for each particle type across the different R_w ranges.

Following the box plot, the angular group (blue) has a distribution range from 0.2 to 0.29, with a median value of 0.25 lying between the reference ranges of the angular and subangular classes. This class has few outliers, but still overlaps with both the subangular and subrounded classes. In addition, the subangular class (green) has a narrow range, varying from 0.24 to 0.31, with the median value falling within the reference range of R_w= [0.25, 0.35]. However, it has many outliers and overlaps with the neighboring angular and subrounded classes. Furthermore, the subrounded group (red) has the widest distribution range and median value within the reference range of R_w= [0.35, 0.49]. Outliers exist, and there is a significant overlap with the reference ranges of the other particle classes. Moreover, the rounded class (purple) has the second-widest distribution, narrower than that of the subrounded class, with a median value within the reference range of R_w= [0.49, 0.7]. However, this class has the most outliers widely distributed, indicating greater variability and higher typical values.

Overall, both plots in Fig. 18 show that the calculated roundness coefficients of the particles generated by the cGAN model are correctly classified with a clear separation between classes with different roundness levels. However, there are still some issues, such as overlap between classes and the presence of outliers. This suggests that the cGAN model can generate diverse data with similar shapes, and that the distribution of the calculated roundness coefficients for the particle class tends to fall within the reference ranges, despite the presence of errors and overlaps. This indicates that the cGAN model generated highly diverse datasets. The calculation of the roundness coefficient can be influenced by several factors, including the presence of noise during data generation. Moreover, the selection of parameters related to the filtering technique, identification of key points on the boundary points of the particle, and calculation of the inscribed tangent circles significantly affect the sensitivity of the roundness coefficient calculation results. However, with concentrated distributions and median values within conventional reference ranges for each particle class, the cGAN model can generate numerous samples to ensure that the distribution of the roundness coefficient is within standard reference ranges. Because of the issues described above when generating ballast particle data using the cGAN model, it is important to verify the roundness

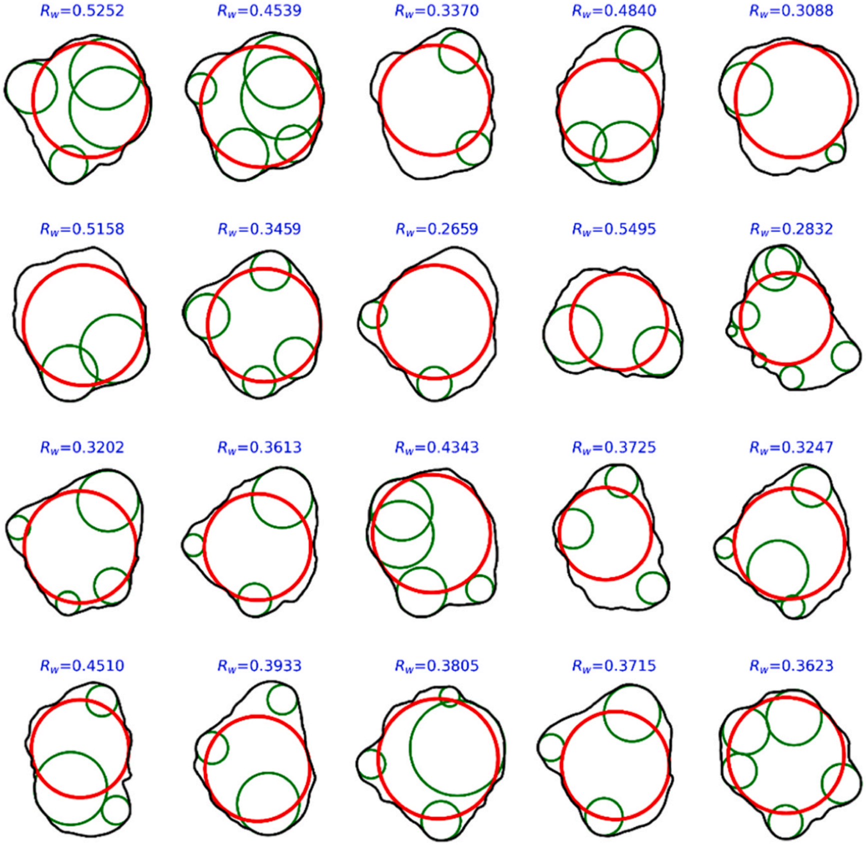


Fig. 16. Roundness calculation for subrounded particles generated by cGAN.

coefficient calculation process to ensure that the generative particles satisfy the conventional roundness coefficient value ranges before using cGAN-generated ballast particles in numerical models such as DEM.

This study developed a cGAN model to generate 2D ballast particles, enhancing the automation and efficiency of data generation for ballasted track simulations. Since railway is typically longitudinal structures, 2D numerical models are commonly used. Moreover, the input data of ballast particles were simplified into numerical vectors of length 360, matching the number of neurons in the input layer of the discriminator and the output layer of the generator in cGAN model. This reduced the complexity of both the discriminator and generator networks. In addition, the cGAN model with an optimal neural network architecture was found to balances the generator and discriminator, leading generated particles have high fidelity and are consistent with the statistical and geometric properties of realistic ballast.

However, there are several limitations in this study. Firstly, the incorporation of the cGAN model into DEM software has not yet been addressed, but the generated ballast particles based on the cGAN model can be used in many applications in future research work, including integration into open-source 2D simulation software such as LAMMPS or YADE. This software may be suitable because it supports custom contact models and allows for shape simplification, such as using polygon or spherical forms. Furthermore, a custom-built tool needs to be developed to convert irregular particles into clump particles that are compatible with each software. Each clump can be made up of several circular particles placed based on local coordinates. According to the list of these smaller particles and the specific requirements of the DEM software, the generated data can be automatically processed and correctly defined within DEM models. Secondly, the cGAN model was focused on 2D ballast particle morphology, considering the irregularities curve. Suppose the ballast particle morphology in simulations does not accurately represent real ballast particles. In that case, the main consequence is a reduction in the reliability and realism of the numerical simulation results, especially when predicting mechanical behavior, performance, and the load-bearing capacity of the ballast layer. In addition, using inaccurate particle shapes can cause errors in important properties such as how particles interact with each other, how loads are spread, and how well particles lock together. These properties are important for understanding the stability, deformation, and wear of ballast under repeated or moving loads. For example, in DEM simulations, using simple or realistic particle shapes can lead to wrong estimates of shear strength, settlement, or how stable the ballasted track is. This can cause discrepancies between simulated and actual performance. Therefore, we proposed a procedure for finding the optimal neural networks of the generator and discriminator, leading to enhanced accurate morphology of the generated ballast particle. Additionally, the cGAN model in this study did not consider the surface texture of the ballast particles, which significantly affects contact interactions and the degradation of the ballasted track. To overcome this issue, an extension of the cGAN

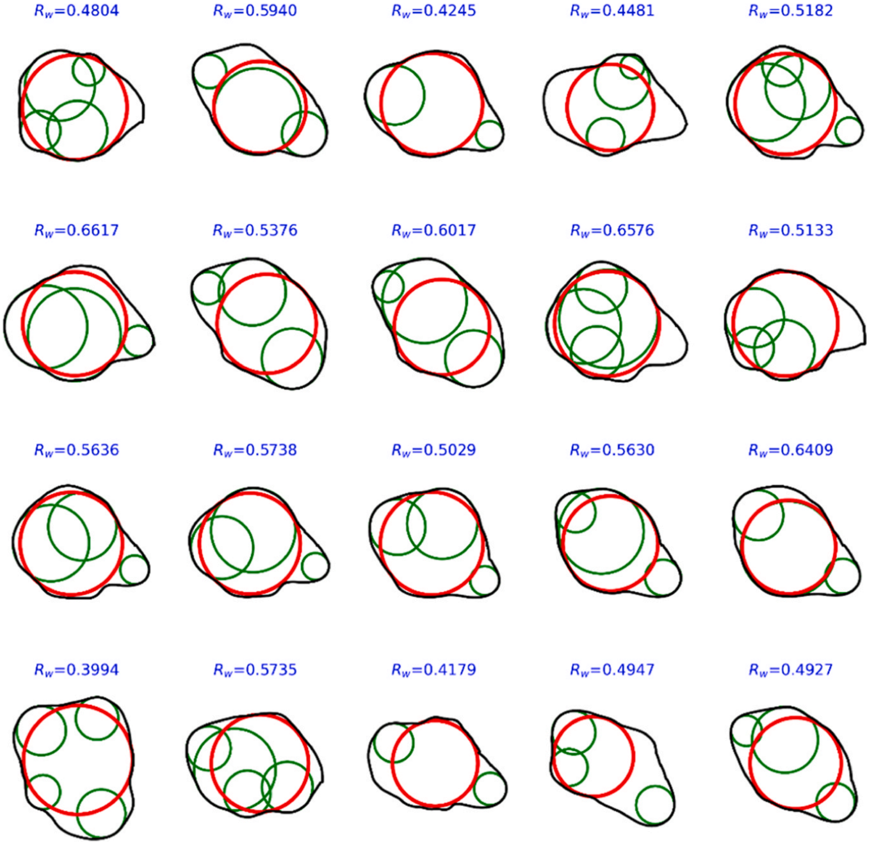


Fig. 17. Roundness calculation for rounded particles generated by cGAN.

can be adapted by combining multiple 2D cGAN models to generate cross-sectional slices of 3D ballast particles, which can then be reconstructed into full 3D shapes using multi-plane synthesis techniques. This approach is possible to improve the texture of the generated ballast particles based on the 2D cGAN model in this study. In future works, the application of generated ballast particle data is not limited to DEM simulations. The dataset generated by the cGAN model can also be used in other areas, such as numerical models in ground-penetrating radar problems to analyze ballast degradation, especially the effective thickness of the ballast layer. Under repeated loading, ballast tends to degrade, reducing the thickness of clean ballast and forming a fouled ballast layer. This process can be analyzed using image processing and artificial intelligence (AI). In the future, the extension of the cGAN model to generate 3D particles may improve its application in more complex DEM models and AI-based degradation analysis of ballasted track using Ground Penetrating Radar, leading to higher accuracy in both simulations and real-world assessments.

5. Summary and conclusions

In this study, we developed a conditional Generative Adversarial Network (cGAN) model to improve the reconstruction of ballast particle morphology for Discrete Element Method (DEM) simulations. Real ballast image data was converted into numerical vectors to facilitate efficient training. The cGAN model increased both the quantity of generated particles with shapes similar to those in the training dataset and enhanced the quality and accuracy of the generated samples, in accordance with standard roundness coefficients.

Key findings include:

1. A ballast class condition is a key input parameter for the cGAN model to generate a specific particle class following the required end-user. This condition categorizes particles into angular, subangular, subrounded, and rounded classes.
2. A key innovation of this study, compared to previous research [35], is the transformation of ballast image data into irregularity particles using numerical vectors composed of 360 elements for training the cGAN model. By converting raw image data in this way, the model can more flexibly and efficiently process and replicate the morphological characteristics of the ballast particles.
3. A proposal of finding optimal neural network architecture of cGAN model was adapted to improve the generator and discriminator networks' stability and effectiveness in generate morphologically accurate ballast particles.
4. The performance of the cGAN model was evaluated using the ROC AUC metric. The resulting values for the angular, subangular, subrounded, and rounded classes were 0.9902, 0.9647, 0.9898, and 0.9862, respectively, with an overall average of 0.9827. These

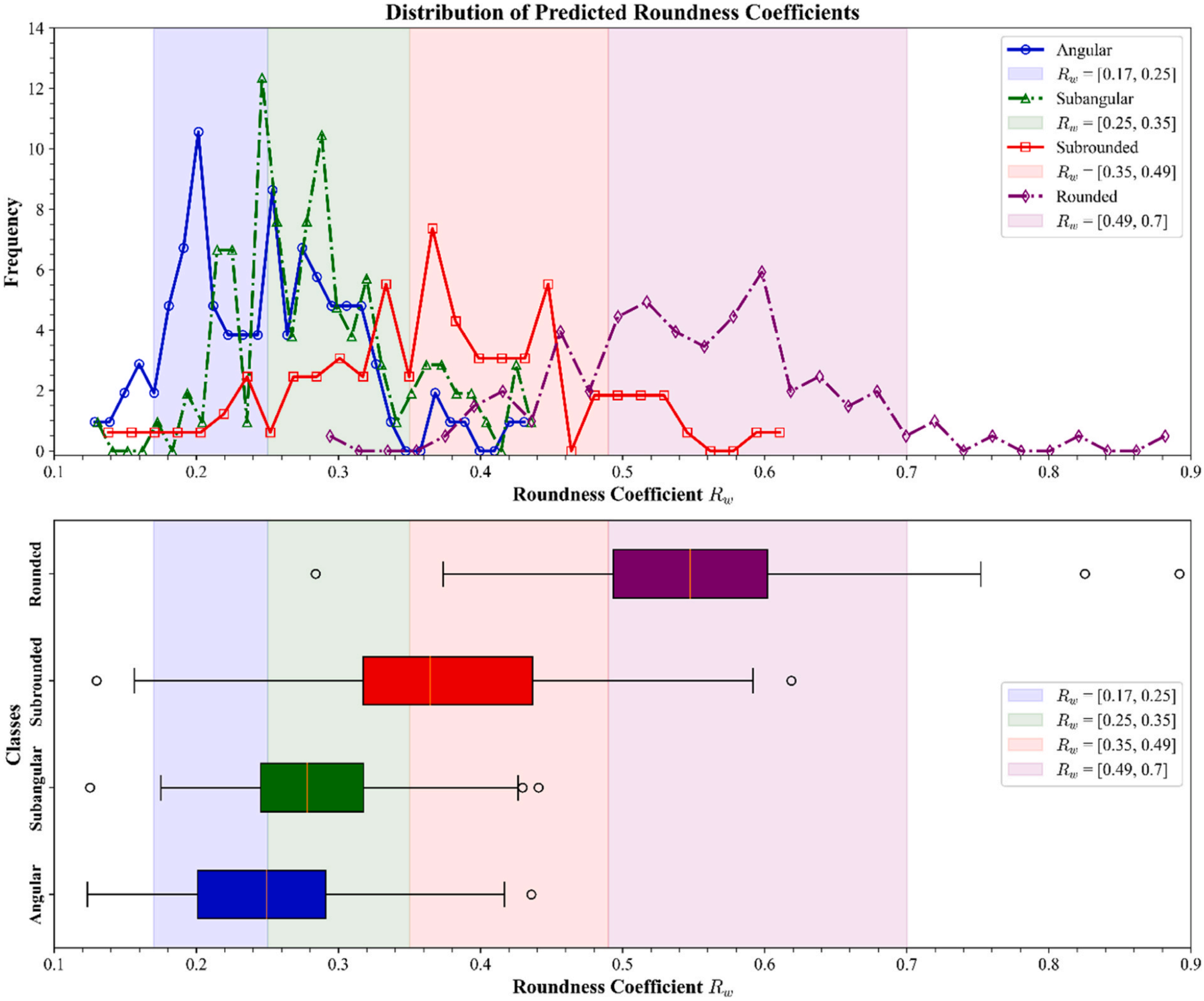


Fig. 18. Distribution of roundness coefficient for different particle classes.

results demonstrated that the cGAN model effectively generated ballast particles with morphological characteristics that closely resemble those of the original ballast particle dataset.

5. The distribution of generated particles from cGAN model were investigated by roundness coefficient. A Python program automatically computed the roundness coefficient was developed through several steps, including noise filtering, identifying key points and corner points, estimating the inscribed tangent circles, and determining the largest inscribed circle. As a result, the roundness coefficients of the generated particles were primarily concentrated within the expected value ranges for each ballast class.

CRediT authorship contribution statement

Gyu-Hyun Go: Writing – review & editing, Funding acquisition, Conceptualization, Supervision, Formal analysis. **Viet Dinh Le:** Validation, Writing – original draft, Software, Conceptualization, Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1C1C1006507).

Code availability

Name of the code/library: KIT-cGAN: Ballast particles generation using cGAN
Contact: vietld@kumoh.ac.kr; (+82)010 2920 1514
Hardware requirements: GPU (e.g., NVIDIA GTX/RTX series or equivalent)
Program language: Python (compatible with versions >= 3.6)
Software required:

- PyTorch > = 1.11
- NumPy > = 1.21
- PyYAML > = 6.0
- Additional dependencies: Matplotlib, pandas, and CSV handling libraries

Program size: Approximately 30 MB (excluding dependencies and dataset)
The source codes are available for downloading at the link: https://github.com/vietld-itec/KIT-cGAN-ballast-particles-generation.

Data availability

Data will be made available on request.

References

[1] P. Aela, L. Zong, W. Powrie, G. Jing, Influence of ballast shoulder width and track superelevation on the lateral resistance of a monoblock sleeper using discrete element method, Transp. Geotech. (2023) https://doi.org/10.101642:101040.
[2] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, Proc. 34th Int. Conf. Mach. Learn. (2017) 214–223.
[3] M. Caudill, Neural networks primer, AI Expert 3 (6) (1988) 53–59.
[4] J. Chen, B. Indraratna, J.S. Vinod, T. Ngo, Y. Liu, Discrete element modelling of the effects of particle angularity on the deformation and degradation behaviour of railway ballast, Transp. Geotech. 43 (2023) 101154, https://doi.org/10.1016/j.trgeo.2023.101154.
[5] L.L. Chun, C.C. Wei, Y. Cheng, Y. Yang, B. Póczos, MMD GAN: Towards deeper understanding of moment matching network, Cornell University Library, Ithaca, 2017, https://doi.org/10.48550/arXiv.1705.08584.
[6] A. Danesh, A.A. Mirghasemi, M. Palassi, Evaluation of particle shape on direct shear mechanical behavior of ballast assembly using discrete element method (DEM), Transp. Geotech. (2020), https://doi.org/10.1016/j.trgeo.2020.100357.
[7] L. Fu, S. Zhou, P. Guo, Z. Tian, Y. Zheng, Dynamic characteristics of multiscale longitudinal stress and particle rotation in ballast track under vertical cyclic loads, Acta Geotech. 16 (5) (2021) 1527–1545, https://doi.org/10.1007/s11440-020-01098-1.
[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (NIPS 2014) 27 (2014) 2672–2680.
[9] Y. Guo, V. Markine, J. Song, G. Jing, Ballast degradation: Effect of particle size and shape using los angeles abrasion test and image analysis, Constr. Build. Mater. 169 (2018) 414–424, https://doi.org/10.1016/j.conbuildmat.2018.02.170.
[10] G. Huang, Learning capability and storage capacity of two-hidden-layer feedforward networks, IEEE Trans. Neural Netw. 14 (2) (2003) 274–281, https://doi.org/10.1109/TNN.2003.809401.
[11] I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, Neurocomputing 10 (3) (1996) 215–236, https://doi.org/10.1016/0925-2312(95)00039-9.
[12] D. Kim, H. Youn, Classifying roundness and sphericity of sand particles using CNN regression models to alleviate data imbalance, Acta Geotech. 19 (2024) 6569–6584, https://doi.org/10.1007/s11440-024-02410-z.
[13] H. Kim, C.T. Haas, A.F. Rauch, C. Browne, 3D image segmentation of aggregates from laser profiling, Comput. Aided Civ. Infrastruct. Eng. 18 (4) (2003) 254–263, https://doi.org/10.1111/1467-8667.00315.
[14] Y. Kim, J. Ma, S.Y. Lim, J.Y. Song, T.S. Yun, Determination of shape parameters of sands: A deep learning approach, Acta Geotech. 17 (4) (2022) 1521–1531, https://doi.org/10.1007/s11440-022-01464-1.
[15] F. Lanaro, P. Tolppanen, 3D characterization of coarse aggregates, Eng. Geol. 65 (1) (2002) 17–30, https://doi.org/10.1016/S0013-7952(01)00133-8.
[16] Y. LeCun, S. Chopra, R. Hadsell, A tutorial on energy-based learning, MIT Press, 2006. ⟨https://yann.lecun.com/exdb/publis/pdf/lecun-06.pdf⟩.
[17] J. Liu, J. Xiao, H. Liu, G. Liu, P. Wang, Y. Lin, Random generation method of ballast 2D topology based on particle characteristics, Constr. Build. Mater. 221 (2019) 762–771, https://doi.org/10.1016/j.conbuildmat.2019.06.131.
[18] Y. Liu, R. Gao, J. Chen, A new DEM model to simulate the abrasion behavior of irregularly-shaped coarse granular aggregates, Granul. Matter 23 (3) (2021), https://doi.org/10.1007/s10035-021-01130-5.
[19] D. Lopez-Paz, M. Oquab, Revisiting classifier two-sample tests, arXiv. Org. (2016), https://doi.org/10.48550/arXiv.1610.06545.
[20] X. Mao, Q. Li, H. Xie, R.Y.K. Lau, Z. Wang, S.P. Smolley, Least squares generative adversarial networks, Pap. Presente (2017) 2813–2821, https://doi.org/10.1109/ICCV.2017.304.
[21] T. Masters, Practical neural network recipes in C++. San Diego. Calif, Acad. Press, 1995.
[22] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv Prepr. (2014), https://doi.org/10.48550/arXiv.1411.1784.
[23] G. Mollon, J. Zhao, 3D generation of realistic granular samples based on random fields theory and Fourier shape descriptors, Comput. Methods Appl. Mech. Eng. 279 (2014) 46–65, https://doi.org/10.1016/j.cma.2014.06.022.
[24] J. Paola, Neural network classification of multispectral imagery, Diss. Univ. Ariz. Tucson (1994).
[25] A.R. Ripley, Statistical aspects of neural networks, Netw. Chaos Stat. Probabalistic Asp. (1993) 40–123.
[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, Cornell University Library, Ithaca, 2016, https://doi.org/10.48550/arXiv.1606.03498.
[27] D. Sejdinovic, A. Gretton, B. Sriperumbudur, K. Fukumizu, Hypothesis testing using pairwise distances and associated kernels. Proceedings of the 29th International Conference on Machine Learning, Edinburgh., Scotland., UK., 2012, https://doi.org/10.48550/arXiv.1205.0411.
[28] C. Shi, Z. Fan, D.P. Connolly, G. Jing, V. Markine, Y. Guo, Railway ballast performance: Recent advances in the understanding of geometry, distribution and degradation, Transp. Geotech. (2023), https://doi.org/10.1016/j.trgeo.2023.101042.
[29] G.J. Székely, Potential and kinetic energy in statistics. Budapest Institute of Technology, Technical University, 1989.
[30] G.J. Székely, E-statistics: The energy of statistical samples. Technical Report., Bowling Green State University, Department of Mathematics and Statistics, 2003.
[31] G.J. Székely, M.L. Rizzo, Testing for equal distributions in high dimension, InterStat (2004) 5.
[32] G.J. Székely, M.L. Rizzo, A new test for multivariate normality, J. Multivar. Anal. 93 (1) (2005) 58–80, https://doi.org/10.1016/j.jmva.2003.12.002.

[33] P. Tahmasebi, Packing of discrete and irregular particles, Comput. Geotech. 100 (2018) 52–61, https://doi.org/10.1016/j.compgeo.2018.03.011.

[34] C. Wang, A theory of generalization in learning machines with neural network applications, Dissertation., University of Pennsylvania, Philadelphia, 1994.

[35] Y. Wang, H. Xiao, Y. Chi, Z. Zhang, Z. Qian, BallastGAN: Random generation of ballast particle contour based on generative adversarial networks, Constr. Build. Mater. (2024), https://doi.org/10.1016/j.conbuildmat.2023.134521.

[36] R. Wettimuny, D. Penumadu, Application of Fourier analysis to digital imaging for particle shape analysis, J. Comput. Civ. Eng. 18 (1) (2004) 2–9, https://doi.org/10.1061/(ASCE)0887-3801(2004)18:1(2).

[37] J. Xiao, D. Zhang, K. Wei, Z. Luo, Shakedown behaviors of railway ballast under cyclic loading, Constr. Build. Mater. 155 (2017) 1206–1214, https://doi.org/10.1016/j.conbuildmat.2017.07.225.

[38] J. Xiao, X. Zhang, D. Zhang, L. Xue, S. Sun, J. Stránský, Y. Wang, Morphological reconstruction method of irregular shaped ballast particles and application in numerical simulation of ballasted track, Transp. Geotech. 24 (2020) 100374, https://doi.org/10.1016/j.trgeo.2020.100374.

[39] H. Yin, Z. Li, J. Zuo, H. Liu, K. Yang, F. Li, Wasserstein generative adversarial network and convolutional neural network (WG-CNN) for bearing fault diagnosis, Math. Probl. Eng. 2020 (2020) 1–16, https://doi.org/10.1155/2020/2604191.

[40] J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial network, Cornell University Library, Ithaca, 2017, https://doi.org/10.48550/arxiv.1609.03126.

[41] J. Zheng, R.D. Hryciw, Traditional soil particle sphericity, roundness and surface roughness by computational geometry, Géotechnique 65 (2015) 494–506, https://doi.org/10.1680/geot.14.P.192.

[42] B. Zhou, J. Wang, H. Wang, Three-dimensional sphericity, roundness and fractal dimension of sand particles, Géotechnique 68 (1) (2018) 18–30, https://doi.org/10.1680/jgeot.16.p.207.