Smart Structures and Systems, *Vol. 27, No. 4 (2021) 607-621*
DOI: https://doi.org/10.12989/sss.2021.27.4.607

Crowd jumping load simulation with generative adversarial networks

Jiecheng Xiong [1] and Jun Chen*[2,3]

[1] *School of Civil Engineering, Zhengzhou University, 100 Kexue Road, Zhengzhou, P.R. China*
[2] *College of Civil Engineering and Architecture, Xinjiang University, Urmuqi, 830047, China*
[3] *State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University, 1239 Siping Road, Shanghai, P.R. China*

(*Received March 7, 2020, Revised December 24, 2020, Accepted January 1, 2021*)

Abstract. To mathematically represent crowd jumping loads, the features of the jumping load of each person, including pulse curve patterns, pulse interval sequences, and pulse energy sequences are considered. These features are essentially high-dimensional random variables. However, they have to be represented in a practically simplified model due to the lack of mathematical tools. The recently emerged generative adversarial networks (GANs) can model high-dimensional random variables well, as demonstrated in image synthesis and text generation. Therefore, this study adopts GANs as a new method for modelling crowd jumping loads. Conditional GANs (CGANs) combined with Wasserstein GANs with gradient penalty (WGANs–GP) are used in pulse curve pattern modelling, where a multi-layer perceptron and convolutional neural network are selected as the discriminator and generator, respectively. For the pulse energy sequence and pulse interval sequence modelling, similar GANs are used, where recurrent neural networks are selected as both the generator and discriminator. Finally, crowd jumping loads can be simulated by connected the pulse samples based on the pulse energy sequence samples and interval sequence samples, generated by the three proposed GANs. The experimental individual and crowd jumping load records are utilized in training GANs to ensure their output can simulate real load records well. Finally, the feasibility of the proposed GANs was verified by comparing the measured structural responses of an existing floor to the predicted structural responses.

Keywords: crowd jumping; human-induced vibration; deep learning; generative adversarial networks

1. Introduction

The structural vibration serviceability and safety problems due to crowd activities have become increasingly significant for designing structures such as long-span floors, pedestrian bridges, and cantilever stands. This is because their natural frequencies and structural damping have reduced with the continuously increasing spans and the establishment of wall-free multifunctional design strategies (Chen *et al*. 2016, Jimenez-Alonso and Saez 2018, Van Nimmen *et al*. 2014, Wang *et al*. 2018). Crowd jumping, compared with other common activities, such as walking, running, and bouncing, causes the largest load on these structures (Xiong and Chen 2018), particularly when the activity of the crowd is coordinated by music or metronome, such as in pop concerts or sports games. Inadequate consideration of crowd jumping loads can result in considerable economic loss and even casualties in rare cases. At a pop concert in London in 1994, a grandstand collapsed due to crowd jumping, injuring 50 people (de Brito and Pimentel 2009). In 2011, the TecoMart building, a 39-story high-rise building with a shopping mall and high-end offices in Seoul, vibrated for 10 min, causing the occupants to escape in panic (Lee *et al*. 2013). The building was closed for two days for field investigation, which caused a total economic loss of 6 billion Won (South-Korean currency), i.e., 5.43 million US dollars. The vibration was subsequently confirmed to have been caused by the exercising of 17 people in the aerobics centre at the 12th floor. This was followed by additional three months of structural safety monitoring using various sensors. The authors are aware of numerous similar jumping-induced incidents in practical engineering which have not been publicly reported for various non-technical reasons.

To assess the vibration performance of a structure subjected to a crowd jumping load, a load model is required either in the design stage or as-built stage (Xiong and Chen 2021). The jumping load is composed of series of pulses, and it is approximately periodic, as displayed in Fig. 5. Therefore, a Fourier series is generally used to model crowd jumping loads in existing studies (Ellis and Ji 2004, Li *et al*. 2018, Parkhouse and Ewins 2006), where the dynamic load factor and the phase angle are obtained by fitting experimental data. The above deterministic models treat crowd jumping loads as a periodic process in which all the jumping pulses are identical. However, crowd jumping loads are stochastic processes, due to the inter-subject randomness (e.g., different people generate various jumping loads) and the intra-subject randomness (e.g., the inability of a person to repeat the same force in each jumping pulse. Ignoring the randomness of crowd jumping loads results in errors in calculating structural responses (Racic and Pavic 2010a). Moreover, because different jumping persons are located at different positions of a structure, each individual force in a crowd has different points of application on the

*Corresponding author, Ph.D., Professor,
E-mail: cejchen@tongji.edu.cn

ISSN: 1738-1584 (Print), 1738-1991 (Online)

occupied structure. Thus, crowd jumping loads are multi-point excitations. The above Fourier series models simplify them as single-point excitations. Therefore, the modal value of each jumping individual cannot be accurately considered when calculating structural responses. This can lead to errors if the modal of the structure is complex (Xiong *et al*. 2021). To this end, researchers have attempted to establish multi-point stochastic models of crowd jumping loads, which can simulate the load time history of each person in the crowd. To develop such a model, the jumping loads must be decomposed into segments of jumping pulses, where the features, such as the pulse curve pattern and the sequence, describing the distribution of the pulses on time axis must be modelled. All these features are essentially high-dimensional random variables, in which high-dimensional random variables refer to multi-dimensional random variables with high dimensionality that can reach tens to thousands. However, they are typically simply modelled in the current practices, due to the lack of appropriate mathematical tools. For example, pre-assumed simple functions, such as cosine squared functions (Sim *et al*. 2008), combined half-sine and half-sine-squared functions (Chen *et al*. 2015a), Gaussian functions (Racic and Pavic 2010b), and wavelet functions (Chen *et al*. 2018), have been adopted to model the curve patterns of the pulses. In sequence modelling, white noise models (Chen *et al*. 2015a), autoregressive models (Chen *et al*. 2018, Sim *et al*. 2008), and auto-power spectral models (Racic and Pavic 2010a) have been adopted. All these pre-assumed functions inevitably introduce additional and artificial assumptions in the model. This yields low-level approximations of the probability distributions of high-dimensional random variables, which results in a large difference between the simulated and real load samples. Except for the model proposed by Sim *et al*. (2008), the above stochastic models can only be used to simulate individual jumping loads. Therefore, it is essential to introduce a new method to model the probability distributions of these jumping load features, establishing a stochastic model to simulate crowd jumping loads.

The deep learning technology offers computational models composed of multiple processing layers to learn the representations of the given data (LeCun *et al*. 2015). These state-of-the-art methods are efficient in identifying the intricate structures in high-dimensional data and have been applied successfully to various domains (Duan *et al*. 2019, Ye *et al*. 2016), such as computer vision, speech recognition, and natural language processing. Among them, generative adversarial networks (GANs) are a class of unsupervised deep learning algorithms, implemented by a system of two neural networks competing by the method of gaming (Goodfellow *et al*. 2014). They can implicitly model the probability distributions of high-dimensional random variables from real samples and use the obtained probability distributions to generate new samples. GANs have been extensively used in various fields, such as three-dimensional (3D) model reconstruction from images (Wu *et al*. 2016) and language models on a character scale (Press *et al*. 2017). Therefore, GANs appear to be suitable for crowd jumping load generation, which has complex data structures.

Motivated by the successful applications of GANs for high-dimensional random variables modelling, this study attempts to employ GANs to simulate crowd jumping loads without any pre-assumed functions. Section 2 briefly introduces GANs and their upgraded versions: Wasserstein GANs with gradient penalty (WGANs-GP) (Gulrajani *et al*. 2017) and conditional GANs (CGANs) (Mirza and Osindero 2014). Sections 3 and 4 present the collection of real samples, including the arrangements of jumping load experiments and the decomposition of measured jumping loads. Section 5 describes the architectures of the proposed GANs and summarizes the hyperparameters employed in them. Section 6 summarises the procedure for simulating crowd jumping loads based on these GANs. The applications of simulated crowd jumping loads are presented in Section 7. The key findings of this study are discussed in Section 8.

2. Generative adversarial networks

2.1 Basic structures of GANs

A typical structure of GANs consists of two parts: a generator $G(\mathbf{z};\theta_g)$ and a discriminator $D(\mathbf{x};\theta_d)$, as displayed in Fig. 1. $G(\mathbf{z};\theta_g)$ is a neural network with parameter θ_g, such as a multilayer perceptron and a convolutional neural network. It adopts noise variables $\mathbf{z}$ with a prior probability distribution p_Z as the input, and its output is the generated sample, $\mathbf{x} \sim p_{\text{model}}$, with the probability distribution p_{model}. Therefore, p_{model} is defined implicitly with p_Z and $G(\mathbf{z};\theta_g)$. $D(\boldsymbol{x};\theta_d)$ is another neural network with parameter θ_d. The input of $D(\mathbf{x};\theta_d)$ is either the real sample, $\mathbf{x} \sim p_{\text{data}}$, or the generated sample, $\mathbf{x} \sim p_{\text{model}}$, where p_{data} is the probability distribution of the real sample. The output of $D(\mathbf{x};\theta_d)$ is a scalar that represents the probability of $\mathbf{x}$ being a real sample.

Because $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$ are both differentiable neural networks, they can be trained simultaneously with the backpropagation algorithm (Rumelhart *et al*. 1986). $D(\mathbf{x};\theta_d)$ is trained to maximize the probability of correctly discriminating between the real and generated samples, whereas $G(\mathbf{z};\theta_g)$ is trained to maximize the probability of $D(\mathbf{x};\theta_d)$ making an error. Therefore, the training process is a mini–max game with the objective of

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[log\, D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_Z}\left[log\left(1 - D\big(G(\mathbf{z})\big)\right)\right] \quad (1)$$

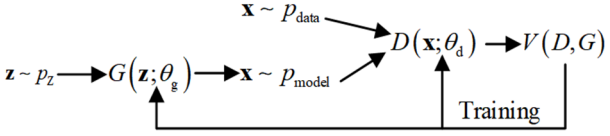


Fig. 1 Typical structure of GANs

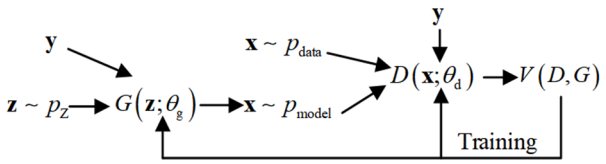


Fig. 2 Typical structure of CGANs

where $\mathbb{E}[\ ]$ represents the expectations. During each training step, parameters θ_g and θ_d are updated. After numerous training steps, if $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$ have sufficient capacity, i.e., in the non-parametric limit, they will reach a global optimum, which is

$$D(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\text{model}}(\mathbf{x})} = \frac{1}{2} \tag{2}$$

In this method, $p_{\text{data}}(\mathbf{x}) = p_{\text{model}}(\mathbf{x})$, i.e., the probability distribution of the generated samples is equal to that of the real samples.

2.2 Conditional GANs

GANs in Section 2.1 cannot implement conditional probability modelling, and there is no control on the modes of the generated samples. To this end, Mirza and Osindero (2014) proposed CGANs, as displayed in Fig. 2.

CGANs introduce conditional variables $\mathbf{y}$ into both $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$. Here, $\mathbf{y}$ can be any type of auxiliary information, such as class labels. The conditioning is performed by feeding $\mathbf{y}$ into both $G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$. Consequently, the objective function of a two-player mini–max game becomes

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[log\, D\,(\mathbf{x}|\mathbf{y})] \\ & + \mathbb{E}_{\mathbf{z}\sim p_Z}\left[log\left(1 - D\big(G(\mathbf{z}|\mathbf{y})\big)\right)\right]\end{aligned} \tag{3}$$

2.3 Wasserstein GANs with gradient penalty

GANs generate highly attractive samples, but they have a problem of unstable and mild training. Arjovsky and Bottou (2017) investigated the reasons for this theoretically, and they identified that GANs in Section 2.1 minimized the Jensen–Shannon (JS) divergence between p_{model} and p_{data}. If p_{data} and p_{model} do not overlap, then their JS divergence will be a constant, log2, so that the gradient is zero. Consequently, if $D(\mathbf{x};\theta_d)$ is trained to optimality before the parameters of the generator are updated, minimizing the JS divergence will lead to vanishing gradients.

To deal with this problem, Gulrajani *et al.* (2017) proposed using WGANs-GP, in which the earth-mover distance between p_{data} and p_{model} is employed instead of the JS divergence. The objective function is redefined as

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_Z}\big[D\big(G(\mathbf{z})\big)\big] \\ & - \lambda \mathbb{E}_{\hat{\mathbf{x}}\sim p_S}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]\end{aligned} \tag{4}$$

where p_S is the probability distribution that is defined implicitly by sampling uniformly along the straight lines between pairs of $\mathbf{x}\sim p_{\text{data}}$ and $\mathbf{x}\sim p_{\text{model}}$, λ is the penalty coefficient, and ∇ is the gradient operation.

3. Experiments for collecting the jumping load records

Real samples of jumping loads are required by the discriminator of GANs. Therefore, a series of experimental campaigns were organized for collecting jumping load records. The force plate is the most widely used human-induced load test instrument, and its accuracy is well recognized. Therefore, it was adopted to measure individual jumping loads. Real samples of high-dimensional random variables related to the load amplitude, e.g., pulses in Section 4.2 and pulse energy sequences in Section 4.3, will be collected by the individual jumping load experiments. However, due to the amount limitations, the loads induced by large crowds cannot be measured using force plates. Thus, 3D motion capture technology (MCT) was adopted to conduct the crowd jumping load experiments. It is an indirect test method of ground reaction forces, in which the displacements of a certain part of the body are measured. Accurate phases of jumping loads can be measured by 3D MCT, which is the key for the modelling of crowd synchronization. Real samples of high-dimensional random variables related to the load phase, e.g., pulse interval sequences in Section 4.4, will be collected by the crowd jumping load experiments, and its accuracy will be verified in Section 4.4.

3.1 Individual jumping load experiment

Fifty-five healthy test subjects participated in the experiments. They were asked to jump to a metronome-guided frequency, which was selected from 1.4 to 2.8 Hz

(a) 1.5 Hz (b) 2.0 Hz

Fig. 3 Individual jumping load records

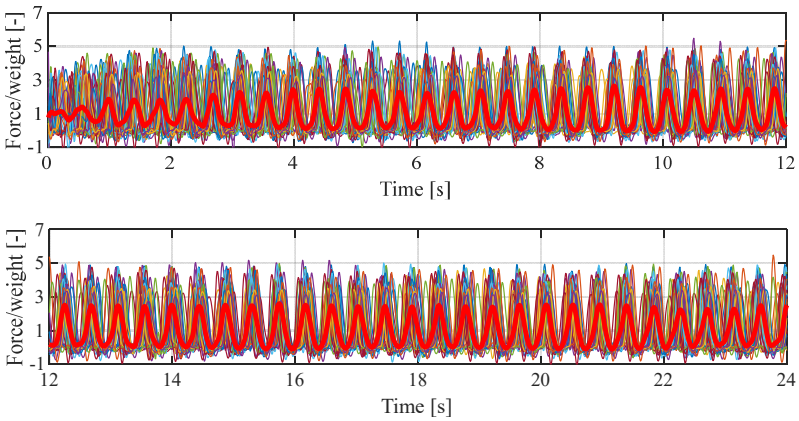


Fig. 4 Crowd jumping loads (thin lines) and global mean (thick line) of the 2.3-Hz test case

with 0.1-Hz intervals, i.e., 15 metronome frequencies were tested. The jumping load was recorded by a force plate (AMTI PB-400600, USA) at a sampling frequency of 100 Hz. The duration of each test case was 25 s. The test protocol satisfied the ethics requirements of Tongji University. Two typical records at 1.5 Hz and 2.0 Hz are depicted in Fig. 3.

3.2 Crowd jumping load experiment by 3D MCT

Ninety-six healthy test subjects participated in the experiment. The experiment was conducted on a rigid ground, and the site was equipped with a motion analysis system (VICON Motion Systems Ltd, UK), which consisted of 18 high-speed infrared cameras. The test subjects were arranged in six rows and eight columns and faced the same direction. They were asked to jump to a metronome-guided frequency, which was randomly selected from 1.5 to 3.5 Hz with 0.1-Hz intervals, i.e., 21 metronome frequencies were tested. Three reflective markers were attached to the clavicle of each test subject. The trajectories of all the markers were recorded by 3D MCT at a sampling frequency of 100 Hz. The test protocol satisfied the ethics requirements of Tongji University. More details regarding the experiment can be found in (Xiong and Chen 2019). The weight-normalized jumping load of the αth test subject, $x_\alpha(t)$, was obtained from the displacement of the corresponding marker, $d_\alpha(t)$, as follows

$$x_\alpha(t) = \ddot{d}_\alpha(t)/\mathrm{g} + 1 \tag{5}$$

where g is the gravity acceleration. The corresponding weight-normalized crowd jumping loads, $\mathbf{x} = \{x_1(t), x_2(t), \cdots, x_n(t)\}^{\mathrm{T}}$, are displayed in Fig. 4. For clarity, each 12-s duration is plotted with the global mean as a thick solid line.

4. Decomposition of recorded samples

Simulation of crowd jumping loads requires modelling the distribution of the pulses on the time axis. Therefore, the pulse curve pattern, pulse interval sequence, and pulse energy sequence must be modelled. The interval sequence describes the arrival time of each pulse, and the energy sequence represents the fluctuations in the pulse energy.

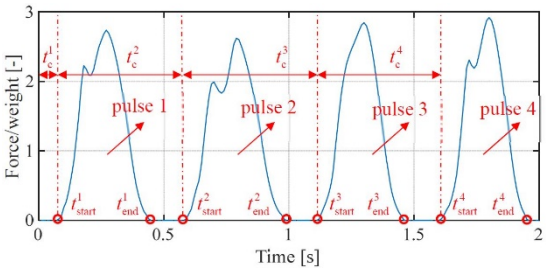


Fig. 5 Decomposition of the jumping loads

4.1 Decomposition of jumping load records

As illustrated in Fig. 5, the jumping loads can be decomposed into a combination of several pulses. The start time, t^k_{start}, and the end time, t^k_{end}, of the kth pulse, $x^k_{\mathrm{pu}}(t)$, are obtained from the point where the ordinate is equal to zero. The pulse interval sequence, $\left(t^1_{\mathrm{c}}, t^2_{\mathrm{c}}, \cdots, t^{n_{\mathrm{pu}}}_{\mathrm{c}}\right)$, is defined as

$$t^k_{\mathrm{c}} = \begin{cases} t^1_{\mathrm{start}} & k = 1 \\ t^k_{\mathrm{start}} - t^{k-1}_{\mathrm{start}} & k > 1 \end{cases} \tag{6}$$

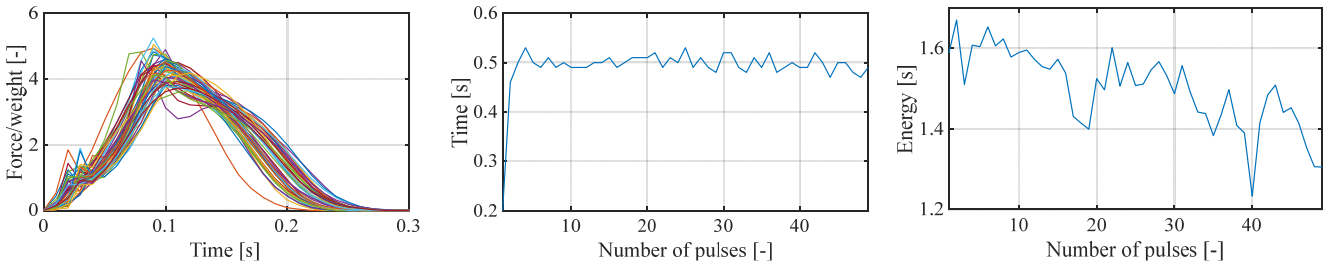
where n_{pu} is the number of pulses. The pulse energy sequence, $(en^1, en^2, \cdots, en^{n_{\mathrm{pu}}})$, is defined as

$$en^k = \int_{t^k_{\mathrm{start}}}^{t^k_{\mathrm{end}}} \left(x^k_{\mathrm{pu}}(t)\right)^2 dt \tag{7}$$

After decomposition, n_{pu} pulses, the pulse interval sequence, and the pulse energy sequence $(en^1, en^2, \cdots, en^{n_{\mathrm{pu}}})$ are obtained. The decomposition results of a 2.0-Hz jumping load are displayed in Fig. 6.

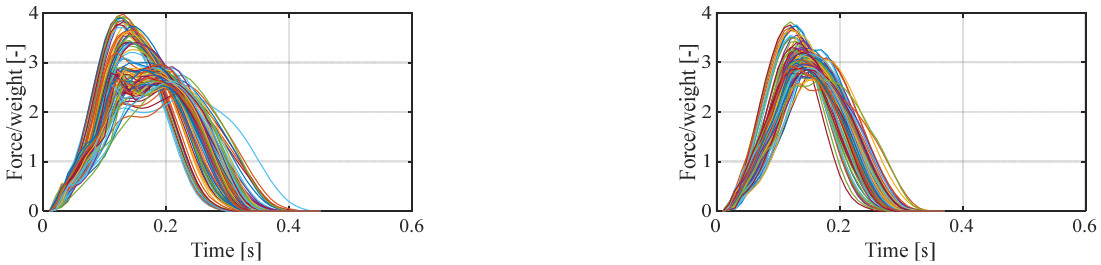
4.2 Pulses

In this study, all the measured individual jumping loads were decomposed based on the above-mentioned procedure. Because a jumping pulse pattern is related to the jumping



(a) Pulses (b) Pulse interval sequence (c) Pulse energy sequence

Fig. 6 Decomposition results of a 2.0-Hz jumping load ($n_{\mathrm{pu}} = 49$)



(a) 1.9 Hz (b) 2.3 Hz

Fig. 7 Real samples of the pulses

(a) 1.9 Hz (b) 2.3 Hz

Fig. 8 Real samples of the pulse energy sequences

frequency (Racic and Pavic 2010b) and not every participant can maintain the metronome frequency (Kasperski and Agu 2005), the collected individual jumping loads were dominant-frequency filtered before decomposition. Fig. 7 depicts the decomposition results of the pulses at 1.9 Hz and 2.3 Hz, where only 100 pieces are illustrated for each jumping frequency for clarity.

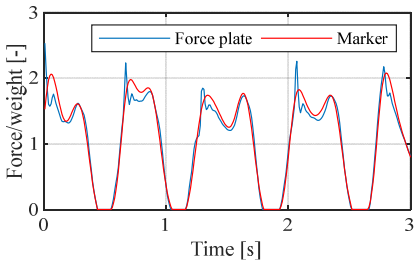
4.3 Pulse energy sequences

In addition to the pulses, the decomposition also provided the pulse energy sequences at different jumping frequencies. Fig. 8 presents samples of the pulse energy sequences at 1.9 Hz and 2.3 Hz, 10 pieces for each jumping frequency, including the first 32 pulses.

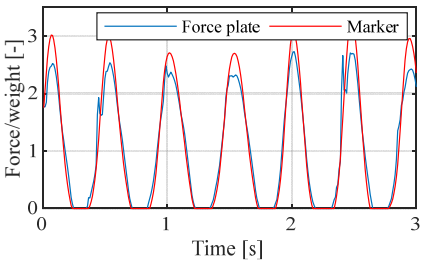
4.4 Pulse interval sequences

The jumping style of a person does not change significantly, irrespective of jumping individually or in a crowd. Therefore, it is reasonable to assume that the samples of the pulses and pulse energy sequences presented in Sections 4.2 and 4.3 can be used to simulate crowd jumping loads. However, a pulse interval sequence from individual jumping load experiments cannot be employed because it does not contain the relative time differences of the different persons in a jumping crowd. To simulate crowd jumping loads, the real samples of the pulse interval sequences have to be acquired from crowd jumping load experiments. Therefore, whether 3D MCT can yield accurate pulse interval sequences must be verified.

Two test subjects participated in the verification experiment. They were asked to jump on force plates (AMTI OR6-7 2000, USA) at four frequencies of 1.5, 2.0, 2.67, and 3.5 Hz, guided by a metronome. One reflective marker was attached to the clavicle of each test subject. The trajectories of the marker and the jumping force of each test subject were simultaneously recorded by a motion analysis system (VICON Motion Systems Ltd, UK). More details about the experiments can be found in (Xiong and Chen 2019). Both the test methods can yield the weight-normalized jumping load, $x_\alpha(t)$, using Eqs. (5) and (8), respectively.



(a) 1.5 Hz



(b) 2.0 Hz

Fig. 9 Comparison of the start times of the pulses

Table 1 Mean difference of the markers and force plates for the pulse interval sequences

Test condition	$t_{c,dif}$ [10^{-2} s]		$\bar{t}_{c,fo}$ [10^{-2} s]	
	Test subject 1	Test subject 2	Test subject 1	Test subject 2
3.5^{1*}	0.72	0.51	28.83	28.49
3.5^{2}	0.68	0.67	30.58	28.60
3.5^{3}	0.59	0.55	29.20	28.67
1.5^{1}	0.78	1.48	66.57	67.06
2.0^{1}	0.85	0.89	50.17	49.98
2.0^{2}	1.35	0.65	49.53	49.72
2.0^{3}	0.60	1.29	50.15	50.04
2.67^{1}	0.46	1.20	36.91	37.40
2.67^{2}	0.70	1.07	37.41	37.41
2.67^{3}	0.45	0.67	37.35	37.35

*3.5^{1} denotes the first case at a 3.5-Hz metronome

$$x_\alpha(t) = F_\alpha(t)/w_\alpha \tag{8}$$

where $F_\alpha(t)$ is the ground reaction force recorded by the force plate and w_α is the weight of the test subject. Subsequently, $x_\alpha(t)$ obtained by the markers was processed using

$$x_\alpha(t) = max\{x_\alpha(t), 0\} \tag{9}$$

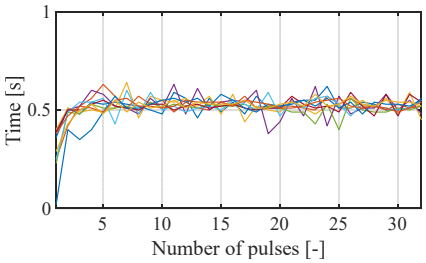
which implies that when $x_\alpha(t)$ is less than 0, it is fixed as 0.

Fig. 9 compares two typical start times of the pulses at 1.5 and 2.0 Hz, and it illustrates that those obtained by the force plates and markers are similar. The pulse interval sequence for each test case in the verification experiment is calculated by Eq. (6). If $\left(t_{c,ma}^1, t_{c,ma}^2, \cdots, t_{c,ma}^{n_{pu}}\right)$ and $\left(t_{c,fo}^1, t_{c,fo}^2, \cdots, t_{c,fo}^{n_{pu}}\right)$ represent the results of the markers and force plates, respectively, the mean difference of two sets of data, $t_{c,dif}$, will be

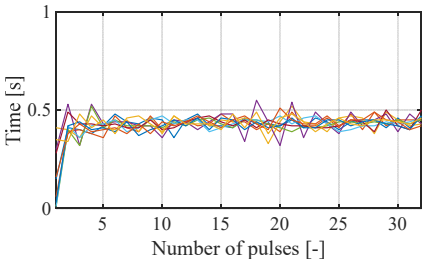
$$t_{c,dif} = \frac{1}{n_{pu}} \sum_{k=1}^{n_{pu}} \left| t_{c,ma}^k - t_{c,fo}^k \right| \tag{10}$$

The mean difference of all the test cases and the average value, $\bar{t}_{c,fo}$, of $\left(t_{c,fo}^1, t_{c,fo}^2, \cdots, t_{c,fo}^{n_{pu}}\right)$ are listed in Table 1. The table shows that the $t_{c,dif}$ values are much smaller than the $\bar{t}_{c,fo}$ values. Therefore, 3D MCT, i.e., crowd jump load experiments as described in Section 3.2, can be used to obtain the real samples of the pulse interval sequences.

Similar to the pulses and pulse energy sequences, the crowd jumping loads were filtered for dominant frequency before decomposition. The filtered records were processed by Eq. (9). All the samples with the same jumping frequency were collected for statistics. Although all the jumping loads of the test subjects were recorded simultaneously in the same test case, the record timings of different test cases possessing the same metronome frequency were asynchronous. To completely utilize all these records to determine the time interval sequence, they must be initially aligned in time. To this end, the 'start time' of the αth test subject is defined as the time instant, $t_{\alpha,start}$, when their $x_\alpha(t)$ first exceeds 1.5 (normalized value, dimensionless). The median value, $t_{start,0.5}$, of all the start times of the test subjects, $\left(t_{1,start}, t_{2,start}, \cdots, t_{n,start}\right)$, in the same test case was calculated. By aligning $t_{start,0.5}$ of



(a) 1.9 Hz



(b) 2.3 Hz

Fig. 10 Real samples of pulse interval sequences

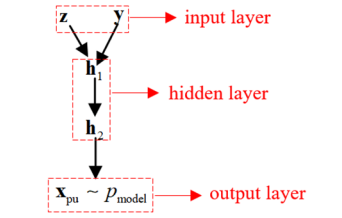
all the test cases with a particular metronome frequency, time synchronization was accomplished. The assumption is that the mean response times of groups jumping to a particular metronome-guided frequency are the same. Subsequently, the pulse interval sequence of each test subject was extracted using the method described in Section 4.1. The real samples of the pulse interval sequences at 1.9 and 2.3 Hz, with 10 pieces for each jumping frequency, including the first 32 pulses, are illustrated in Fig. 10.

5. Architectures and hyperparameters

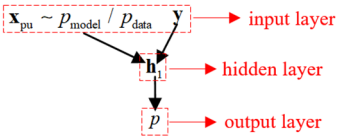
After obtaining the real samples, as detailed in Section 4, three GANs were developed for modelling the jumping pulses, pulse energy sequences, and pulse interval sequences, respectively. A crowd can jump cooperatively from 1.5 to 2.8 Hz, so that features in the above frequency range were modelled, as described in this section. The computational details of these GANs, including the architecture of generator $G(\mathbf{z};\theta_g)$, the architecture of discriminator $D(\mathbf{x};\theta_d)$, optimization objective function $V(D,G)$, and optimization algorithms, are also described here.

5.1 Pulse generation

Discretization of t transforms the modelling of $\mathbf{x}_{pu}(t)$ into the modelling of $\mathbf{x}_{pu}$, and each $\mathbf{x}_{pu}$ is treated as an independent sample of the high-dimensional random variable, $\mathbf{X}_{pu}$. Subsequently, GANs are adopted to model the probability distribution of $\mathbf{X}_{pu}$, which is utilized for generating new pulses.



(a) Generator



(b) Discriminator

Fig. 11 Architectures of the generator and the discriminator

5.1.1 Optimization objective function

As displayed in Fig. 7, the curve pattern of a pulse is related to the jumping frequency, indicating that CGANs that provide additional label information are required. Moreover, the WGANs-GP can improve the training stability and generate high-quality samples. Therefore, the corresponding $V(D,G)$ is a combination of CGANs and WGANs-GP, i.e.

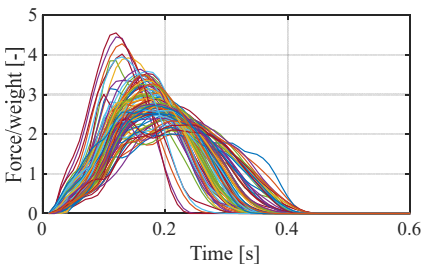
$$\begin{aligned}&\min_G \max_D V(D,G)\\&=\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[D(\mathbf{x}|\mathbf{y})]-\mathbb{E}_{\mathbf{z}\sim p_Z}\left[D\left(G(\mathbf{z}|\mathbf{y})\right)\right]\\&-\lambda\mathbb{E}_{\hat{\mathbf{x}}\sim p_S}[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}}|\mathbf{y})\|_2-1)^2]\end{aligned} \tag{11}$$

5.1.2 Architectures of generator and discriminator

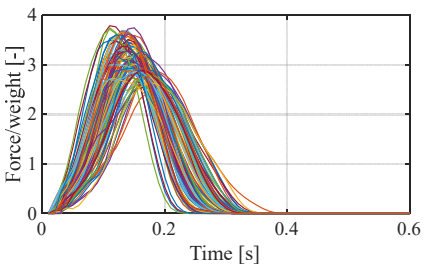
$G(\mathbf{z};\theta_g)$ and $D(\mathbf{x};\theta_d)$ adopt a convolutional neural network and a multilayer perceptron, respectively. These networks require the length of input layer, i.e., the length of real samples, to be consistent. To satisfy this requirement, each real pulse sample, obtained in Section 4.2, was padded with zero at its end to have an identical length as 100.

The architecture of $G(\mathbf{z};\theta_g)$ is displayed in Fig. 11(a). Its input layer includes a noise variable $\mathbf{z}\sim p_Z$ and a condition variable $\mathbf{y}$. $\mathbf{z}$ is a 100-dimensional random variable, where each unit is independent and follows the uniform distribution, $U(0,1)$. $\mathbf{y}$ is a one-hot vector to represent the pulse category. The following are two fully connected hidden layers, $\mathbf{h}_1$ and $\mathbf{h}_2$ whose lengths are 100. The output, i.e., generated sample $\mathbf{x}_{pu}\sim p_{\text{model}}$, is a one-dimensional convolutional layer, whose kernel size, stride, input channel number, and output channel number are 10, 1, 1, and 1, respectively. The activation function for $\mathbf{h}_1$, $\mathbf{h}_2$, and $\mathbf{x}_{pu}\sim p_{\text{model}}$ is rectified linear unit (ReLU) (Krizhevsky *et al*. 2012).

The architecture of $D(\mathbf{x};\theta_d)$ is depicted in Fig. 11(b). Its input layer includes a real sample, $\mathbf{x}_{pu}\sim p_{\text{data}}$, or a generated sample, $\mathbf{x}_{pu}\sim p_{\text{model}}$, and $\mathbf{y}$. The following is a fully connected hidden layer with 100 units, and its activation function is ReLU. The output layer, p, is a scalar indicating the probability that the input sample is real, which adopts a linear activation.



(a) 1.9 Hz



(b) 2.3 Hz

Fig. 12 Generated samples of the pulses

5.1.3 Training process

$V(D,G)$ in Eq. (11) was trained by Adam algorithm (Kingma and Ba 2014), which is the most commonly used one in the backpropagation method. The learning rate, first moment estimation, and second moment estimation were set as 0.0001, 0.5, and 0.999, respectively. One million training steps with a batch size of 128 were conducted. In each step, $D(\mathbf{x};\theta_\mathrm{d})$ was optimized five times and $G(\mathbf{z};\theta_\mathrm{g})$ was optimized once.

5.1.4 Generated samples

After the training, $G(\mathbf{z};\theta_\mathrm{g})$ can generate pulse samples using $\mathbf{z}$ and $\mathbf{y}$. Fig. 12 presents the generated samples of the pulses at 1.9 and 2.3 Hz, with 100 pieces for each frequency. It can be noted from Fig. 12 that the waveform of the generated samples is extremely similar to that of the real samples in Fig. 7 in aspects such as the time history, curve pattern, and peak value.

5.2 Pulse energy sequence generation

Similar to $\mathbf{x}_\mathrm{pu}$, each $\mathbf{en}=(en^1,en^2,\cdots,en^{n_\mathrm{pu}})$ is treated as an independent sample of the high-dimensional random variable, $\mathbf{EN}=(EN^1,EN^2,\cdots,EN^{n_\mathrm{pu}})$ Subsequently, GANs are adopted for modelling the probability distribution of $\mathbf{EN}$, which is utilized to generate new pulse energy sequences.

5.2.1 Optimization objective function

Pulse energy sequence generation is similar to pulse generation. The type of generated samples should be controlled, and WGANs-GP are adopted to increase the training stability and quality of the generated samples. Therefore, it also uses Eq. (11) as the objective function.

5.2.2 Architectures of generator and discriminator

Unlike pulse generation, the length of a pulse energy sequence cannot be limited to generate crowd jumping loads of any duration. Therefore, both $G(\mathbf{z};\theta_\mathrm{g})$ and $D(\mathbf{x};\theta_\mathrm{d})$ adopt recurrent neural networks.

The architecture of $G(\mathbf{z};\theta_\mathrm{g})$ is depicted in Fig. 13(a). The hidden layer, $\mathbf{h}$, adopts a gate recurrent unit (GRU) cell (Cho *et al*. 2014). The initial state of $\mathbf{h}$ consists of $\mathbf{z}_1$ and $\mathbf{y}$. $\mathbf{z}_1$ is a 60-dimensional random variable, where each unit is independent and follows the normal distribution, $N(0,10^2)$. $\mathbf{y}$ is a one-hot vector representing the category of the pulse energy sequence. The first step input of $G(\mathbf{z};\theta_\mathrm{g})$ is $\mathbf{z}_2$, which follows the normal distribution, $N(0,1^2)$. The fully connected layer with the linear activation function is added to the hidden layer of the kth state, $\mathbf{h}^k$, to obtain the kth step output, en^k. The input of step k+1 is equal to the output of step k. A generated sample can be obtained by concatenating the output of each step.

The architecture of $D(\mathbf{x};\theta_\mathrm{d})$ is presented in Fig. 13(b). Its hidden layer h also adopts a GRU cell. $\mathbf{h}$ contains 60 units, and its initial state $\mathbf{h}^0$ is $\mathbf{0}$. The input of $D(\mathbf{x};\theta_\mathrm{d})$ contains $\mathbf{y}$ and $\mathbf{en}\sim p_\mathrm{model}$ or $\mathbf{en}\sim p_\mathrm{data}$. y^k is the kth unit of $\mathbf{y}$, and n_c is the length of $\mathbf{y}$. A fully connected layer with a linear activation function is added to the final state, $\mathbf{h}^{n_\mathrm{c}+n_\mathrm{pu}}$, to obtain the output, p, of $D(\mathbf{x};\theta_\mathrm{d})$.

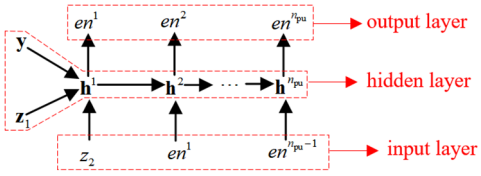
5.2.3 Training process

The training of the pulse energy sequences utilizes the curriculum learning method (Bengio *et al*. 2009). It continuously increases the length of the sample during the training process. In the first training stage, the length of the samples is one, i.e., n_pu in Fig. 13 is equal to one. Then, n_pu is gradually increased from 1 to 32 in intervals of 1. The values of the parameters obtained in the ith stage training are adopted as the initial values of the parameter at the (i+1)th stage training.

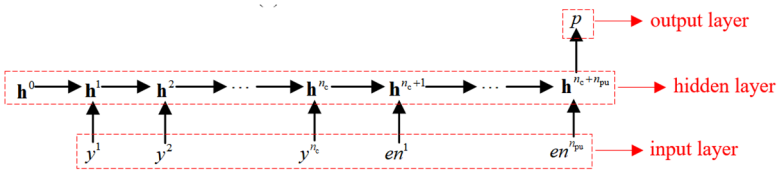
At each training stage, $V(D,G)$ is trained by Adam algorithm. The learning rate, first moment estimation, and second moment estimation were set as 0.0001, 0.5, and 0.9, respectively. A total of 20,000 training steps with a batch size of 64 were conducted. $D(\mathbf{x};\theta_\mathrm{d})$ was optimized five times, and $G(\mathbf{z};\theta_\mathrm{g})$ was optimized once in each training step.

5.2.4 Generated samples

After training, $G(\mathbf{z};\theta_\mathrm{g})$ can generate the pulse energy sequence samples using $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{y}$. Note that although the maximum lengths of the samples during training are 32

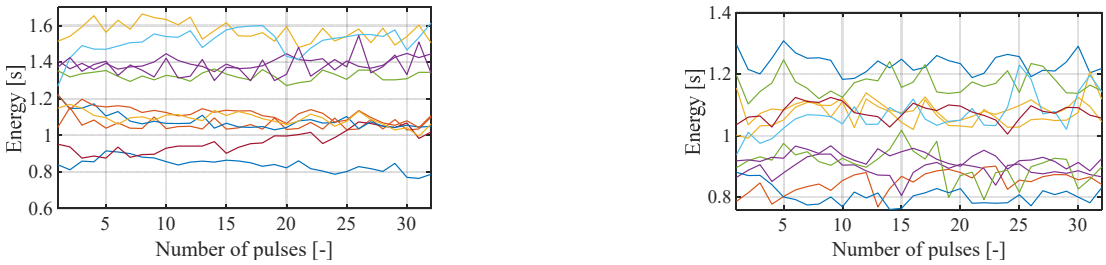


(a) Generator



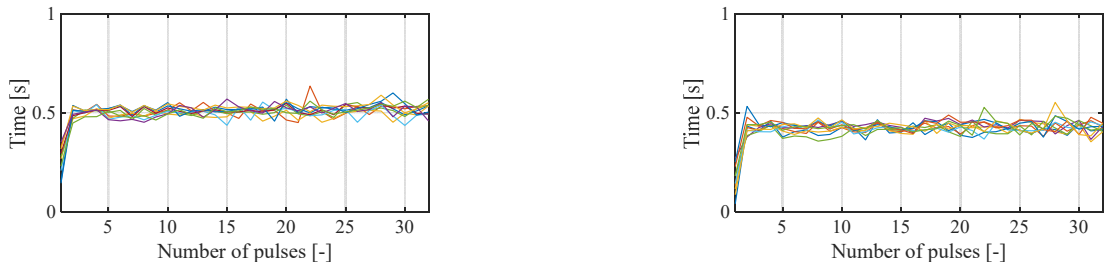
(b) Discriminator

Fig. 13 Architectures of the generator and the discriminator



(a) 1.9 Hz (b) 2.3 Hz

Fig. 14 Generated samples of the pulse energy sequences



(a) 1.9 Hz (b) 2.3 Hz

Fig. 15 Generated samples of the pulse interval sequences

each, the lengths of the generated samples are not limited after the training. Fig. 14 displays the generated samples of the pulse energy sequences at 1.9 and 2.3 Hz, with 10 pieces for each jumping frequency, containing the first 32 pulses. Note from Fig. 14, the waveform of the generated samples is extremely similar to that of the real samples of Fig. 8.

5.3 Pulse interval sequence generation

Similar to $\mathbf{x}_{\mathrm{pu}}$ and $\mathbf{en}$, each $\mathbf{t}_{\mathrm{c}} = \left(t_{\mathrm{c}}^1, t_{\mathrm{c}}^2, \cdots, t_{\mathrm{c}}^{n_{\mathrm{pu}}}\right)$ is treated as an independent sample of the high-dimensional random variable, $\mathbf{T}_{\mathrm{c}} = \left(T_{\mathrm{c}}^1, T_{\mathrm{c}}^2, \cdots, T_{\mathrm{c}}^{n_{\mathrm{pu}}}\right)$. Subsequently, GANs are adopted for modelling the probability distribution of $\mathbf{T}_{\mathrm{c}}$, which is utilized to generate new pulse interval sequences.

The data structures of the pulse energy sequence and pulse interval sequence are identical although the specific values are different. Therefore, the optimization objective function, architectures of the generator and discriminator, and training process are the same as those used in the pulse energy sequence modelling. Fig. 15 presents the generated samples of the pulse interval sequences at 1.9 and 2.3 Hz, with 10 pieces for each jumping frequency, and containing the first 32 pulses. It is noted from Fig. 15 that the waveform of the generated samples is highly similar to that of the real samples of Fig. 10.

6. Procedure for simulating crowd jumping loads

Suppose that a crowd is jumping at a given frequency, e.g., guided by sound stimuli (such as that provided by a metronome or music), then commonly, some people can follow the beat, whereas the others cannot (Kasperski and Agu 2005). The probability distribution of the jumping frequency for the different metronome frequencies is provided in the appendix, and the detailed source information can be found in the literature (Xiong and Chen 2019).

The samples of the pulses, pulse energy sequences, and pulse interval sequences can be generated using the GANs in Section 5. The applicable jumping frequency is in the range of 1.5-2.8 Hz with a 0.1-Hz interval. The weight-normalized crowd jumping loads, $\mathbf{x} = \{x_1(t), x_2(t), \cdots, x_n(t)\}^{\mathrm{T}}$, can be simulated by the following six steps:

- The jumping frequency, $f_{\mathrm{jump},\alpha}$ of the αth person is randomly generated from the probability distribution provided in Table A1, and $f_{\mathrm{jump},\alpha}$ is rounded to the decimal point.
- Numerous pulse samples $\mathbf{x}_{\mathrm{pu}}$ are generated for 1.5-2.8 Hz with a 0.1 Hz interval using the generator developed in Section 5.1.
- The number of pulses, $n_{\mathrm{pu},\alpha}$, required by the $x_\alpha(t)$ of the αth person is estimated by the duration of $x_\alpha(t)$ and $f_{\mathrm{jump},\alpha}$.
- $\left(t_{\mathrm{c}}^1, t_{\mathrm{c}}^2, \cdots, t_{\mathrm{c}}^{n_{\mathrm{pu},\alpha}}\right)$ and $(en^1, en^2, \cdots, en^{n_{\mathrm{pu},\alpha}})$ of the αth person are generated by the generator described in Sections 5.2 and 5.3 based on $f_{\mathrm{jump},\alpha}$ and $n_{\mathrm{pu},\alpha}$.
- According to en^k of the αth person and $f_{\mathrm{jump},\alpha}$, $\mathbf{x}_{\mathrm{pu}}$ generated in the second procedure, whose energy is closest to en^k, is identified as the kth step $\mathbf{x}_{\mathrm{pu}}^k$.
- $\mathbf{x}_{\mathrm{pu}}^k$ of the αth person is connected according to their $\left(t_{\mathrm{c}}^1, t_{\mathrm{c}}^2, \cdots, t_{\mathrm{c}}^{n_{\mathrm{pu},\alpha}}\right)$ to obtain $x_\alpha(t)$.

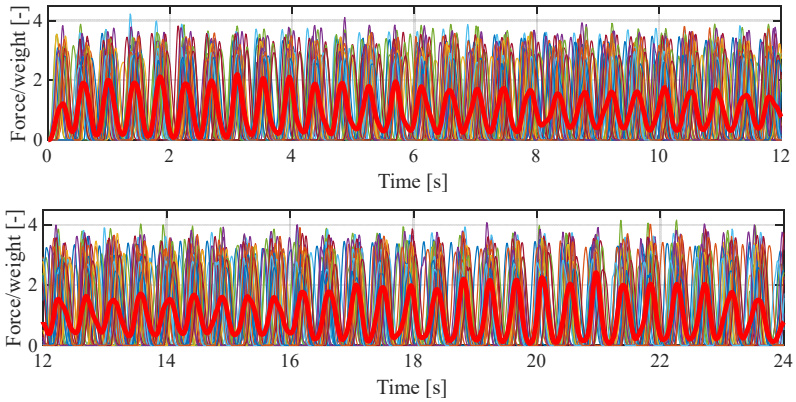


Fig. 16 Crowd jumping loads simulated by GANs

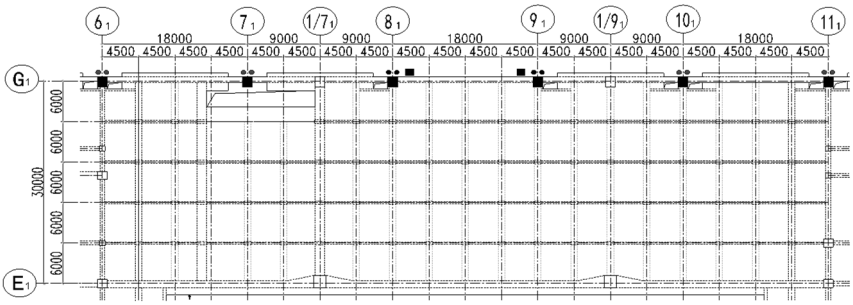
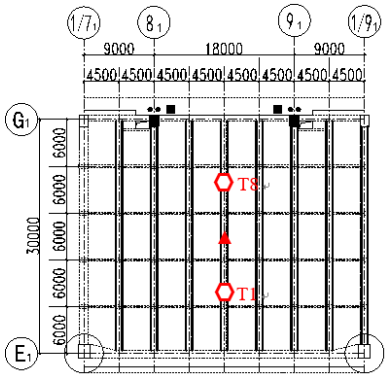
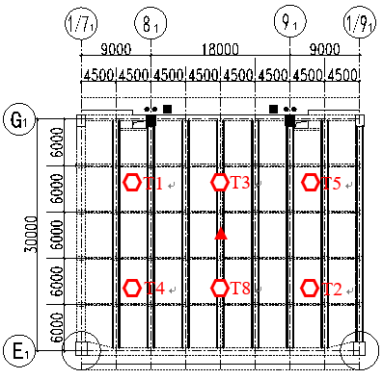


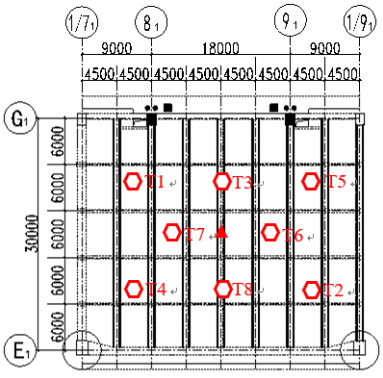
Fig. 17 Plan layout of the test floor



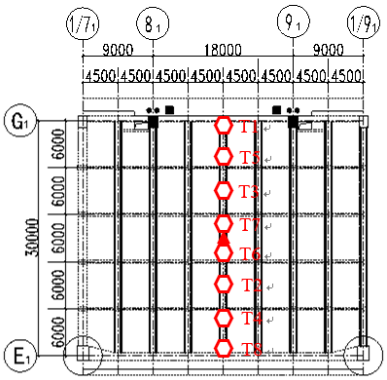
(a) Test case 1



(b) Test case 2



(c) Test case 3



(d) Test case 4

Fig. 18 Locations of the test subjects and the accelerometer

Fig. 16 presents the simulated crowd jumping loads at 2.3 Hz by the above six steps. The simulation includes 30 persons, and its duration is 24 s. The thick solid line is the mean of the crowd jumping loads.

7. Applications

Due to its stochastic nature, experimental verification is not expected to provide a comprehensive proof. However, a limited validation of the data from controlled crowd jumping tests may be effective for this purpose. In this regard, field tests of an existing long-span floor subjected to controlled crowd jumping were conducted. The floor is the waiting hall of a train station in a city in China. It is formed of concrete with a 400-mm thickness and utilizes an external prestressing structural system to achieve a 30-m long span, as depicted in Fig. 17.

Eight test subjects participated in the experiment. They were identified as T1–T8, and their body weights were 68.1, 83.0, 65.0, 58.0, 55.9, 53.5, 53.0, and 67.0 kg, respectively. The experiment contained four test cases, and the locations of the test subjects are illustrated in Fig. 18. They jumped at 2.3 Hz, guided by a metronome, to generate resonance responses. An accelerometer was installed on the floor centre to record the acceleration responses at a sampling frequency of 200 Hz. Its location is plotted in Fig. 18 as a red triangle. More details about the floor and the field tests can be found in (Chen *et al*. 2015b).

The measured fundamental natural frequency of the floor is 2.2 Hz. The damping ratio of the first vibration mode is adopted as 0.03, and the damping ratio of the rest vibration modes is adopted as 0.015. Then, a finite element model of the floor was developed using Midas Gen software. The natural frequency and modal mass calculated by the model are listed in Table 2. The fundamental natural frequency agrees with the measurement. Therefore, the modal properties calculated by the model are adopted for the structural response simulations.

Structural acceleration responses $a(t)$ are calculated by the Newmark-beta method, where the crowd jumping loads, $\mathbf{x} = \{x_1(t),\ x_2(t), \cdots, x_n(t)\}^{\mathrm{T}}$, are simulated by the procedure mentioned in Section 6. Because $\mathbf{x} = \{x_1(t),\ x_2(t), \cdots, x_n(t)\}^{\mathrm{T}}$ is randomly generated, $a(t)$, its peak value a_{peak} and peak 10-s running root-mean-square value, $a_{\mathrm{rms,10s}}$, are random. For each test case, 200 simulations are performed, and the mean, 5%, and 95% of simulated a_{peak} and $a_{\mathrm{rms,10s}}$ are compared with the measurements, as presented in Table 3. Moreover, three simulated $a(t)$ are randomly selected and compared with the measurements, as illustrated in Fig. 19.

8. Conclusions

In this study, GANs provide a novel tool to learn the probability distributions of high-dimensional random variables from given samples and utilize the distribution to generate new samples. In this regard, the crowd jumping loads are decomposed into pulses, pulse energy sequences, and pulse interval sequences. They are all treated as high-dimensional random variables and modelled by GANs. It alters the modeling paradigm of conventional methods, in which the jumping load features are often simplified to independent random variables through artificial assumptions. The generated samples are found to have similar waveforms to those of the real samples. The crowd jumping loads can be simulated by connecting the generated pulses based on the generated pulse interval sequences and energy sequences.

Even better, two applications and extensions of the proposed model are possible:

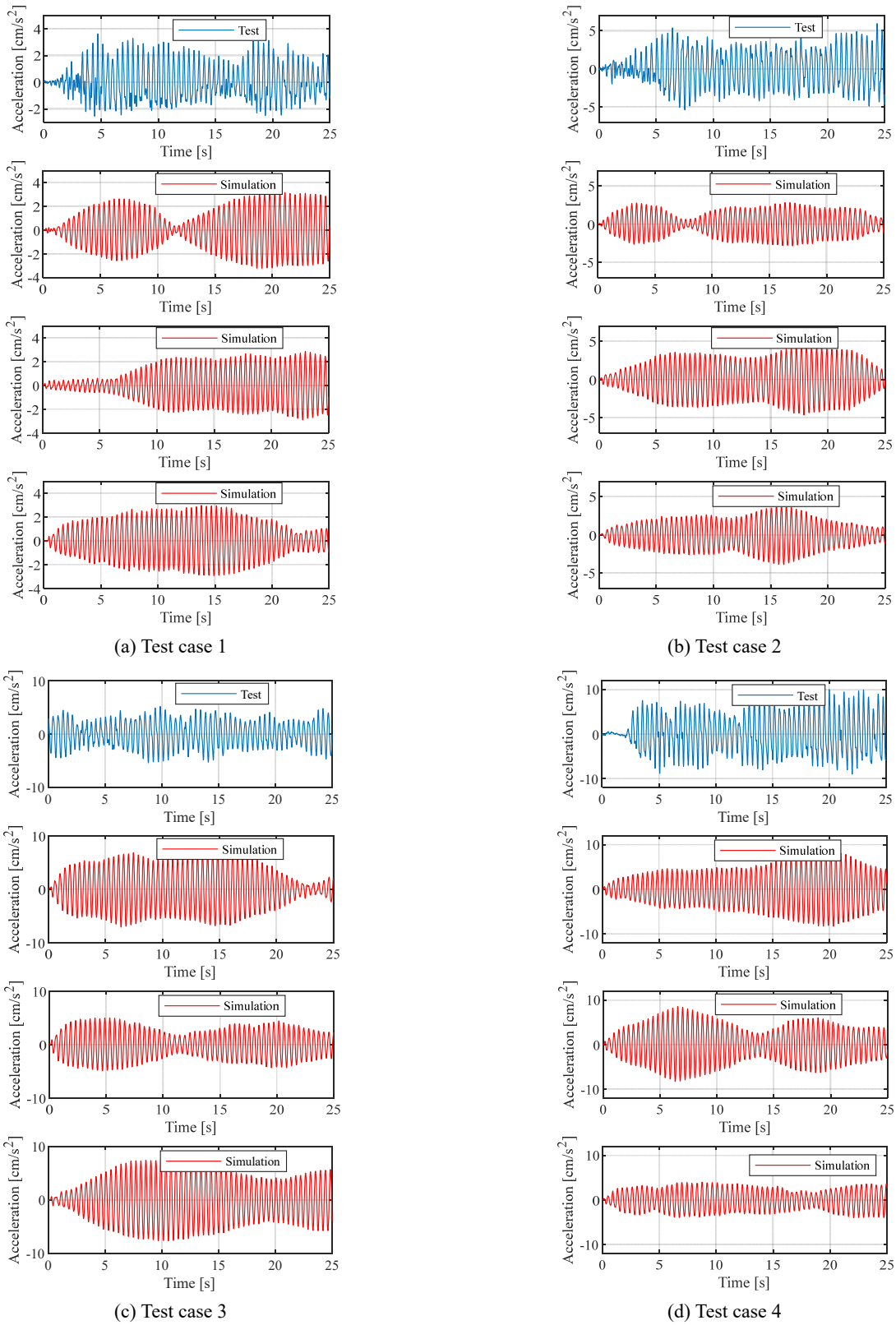
- The simulated crowd jumping loads are applicable to calculate structural responses using Monte Carlo simulations, which was verified by comparing the simulations with the measurements from a real floor. It enhances the evolution of vibration performance for structures dynamically excited by jumping crowds.
- Other dynamic excitations in civil engineering, such as earthquake excitations, fluctuating wind loads and other types of human-induced loads, have similar data structures to crowd jumping loads. Modelling

Table 2 Modal properties obtained by the finite element model

Mode number	1	2	3	4	5	6	7	8	9	10
Frequency [Hz]	2.256	2.666	3.011	3.801	4.220	4.621	4.793	4.797	4.932	5.652
Mass [10^5 kg]	7.425	9.069	7.429	8.115	5.325	5.649	4.504	5.514	4.355	7.150

Table 3 Measured a_{peak} and $a_{\mathrm{rms,10s}}$ and the corresponding simulation results

Test case	1		2		3		4	
	a_{peak}	$a_{\mathrm{rms,10s}}$	a_{peak}	$a_{\mathrm{rms,10s}}$	a_{peak}	$a_{\mathrm{rms,10s}}$	a_{peak}	$a_{\mathrm{rms,10s}}$
Measurements [cm/s^2]	3.61	1.43	5.96	2.63	5.16	2.63	9.96	5.05
Mean simulation [cm/s^2]	2.79	1.69	4.07	2.37	6.62	3.84	6.06	3.44
5% simulation [cm/s^2]	1.87	0.97	2.54	1.36	3.66	1.98	3.66	1.87
95% simulation [cm/s^2]	3.48	2.28	5.68	3.59	10.0	6.16	9.12	5.49



(a) Test case 1

(b) Test case 2

(c) Test case 3

(d) Test case 4

Fig. 19 Comparison of the measured and simulated $a(t)$

these dynamic excitations frequently encounters the problem of dealing with high-dimensional random variables. The proposed approach is demonstrated to be a promising method to solve this problem. Moreover, the structures of the GANs established in this paper can provide reference for future dynamic load modelling.

The limitations of this work can be mainly manifested in the following two points:

- It requires time history analysis and Monte Carlo simulations to conduct assessment of vibration performance for structures, which is time consuming and limits the use of crowd jumping load models. In future study, design values of crowd jumping loads of different reliabilities will be determined through a large number of simulations of crowd jumping loads.
- The real samples are collected from individual and crowd jumping load experiments on a rigid ground, which ignores the effect of structural vibrations on jumping motions. Therefore, the established load model may not be suitable for calculating structural vibrations that are too much perceptible to jumping crowds. In future study, jumping load experiments on flexible structures will be conducted to enrich the real samples.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (51778465, U1711264, 5200837) and Project ZYJKFW201811009. Moreover, the authors would like to thank all the test subjects for participating in the project and making the data collection possible.

References

Arjovsky, M. and Bottou, L. (2017), "Towards principled methods for training generative adversarial networks", *arXiv*, 1701.04862.

Bengio, Y., Louradour, J., Collobert, R. and Weston, J. (2009), "Curriculum learning", *Proceedings of the 26th Annual International Conference on Machine Learning*, Quebec, Canada, June.

Chen, J., Wang, H. and Wang, L. (2015a), "Experimental investigation on single person's jumping load model", *Earthq. Eng. Eng. Vib.*, **14**(4), 703-714. https://doi.org/10.1007/s11803-015-0055-9

Chen, J., Zhang, M. and Liu, W. (2015b), "Vibration serviceability performance of an externally prestressed concrete floor during daily use and under controlled human activities", *J. Perform Constr. Fac.*, **30**(2), 4015007. https://doi.org/10.1061/(asce)cf.1943-5509.0000738

Chen, J., Tan, H. and Pan, Z. (2016), "Experimental validation of smartphones for measuring human-induced loads", *Smart Struct. Syst.*, *Int. J.*, **18**(3), 625-642. https://doi.org/10.12989/sss.2016.18.3.625

Chen, J., Li, G. and Racic, V. (2018), "A data-driven wavelet-based approach for generating jumping loads", *Mech. Syst. Signal Pr.*, **106**, 49-61. https://doi.org/10.1016/j.ymssp.2017.12.013

Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y. (2014), "On the properties of neural machine translation: Encoder-decoder approaches", *arXiv*, 1409.1259.

de Brito, V.L. and Pimentel, R.L. (2009), "Cases of collapse of demountable grandstands", *J. Perform Constr. Fac.*, **23**(3), 151-59. https://doi.org/10.1061/(asce)cf.1943-5509.0000006

Duan, Y., Chen, Q., Zhang, H., Yun, C.B., Wu, S. and Zhu, Q. (2019), "CNN-based damage identification method of tied-arch bridge using spatial-spectral information", *Smart Struct. Syst.*, *Int. J.*, **23**(5), 507-520. https://doi.org/10.12989/sss.2019.23.5.507

Ellis, B.R. and Ji, T. (2004), "Loads generated by jumping crowds: Numerical modelling", *Struct. Eng.*, **82**(17), 35-40.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), "Generative adversarial nets", *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Quebec, Canada, December.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C. (2017), "Improved training of wasserstein gans", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, CA, USA, December.

Jimenez-Alonso, J.F. and Saez, A. (2018), "Motion-based design of TMD for vibrating footbridges under uncertainty conditions", *Smart Struct. Syst.*, *Int. J.*, **21**(6), 727-740. https://doi.org/10.12989/sss.2018.21.6.727

Kasperski, M. and Agu, E. (2005), "Prediction of crowd-induced vibrations via simulation", *Proceeding of 23rd International Modal Analysis Conference*, FL, USA, February.

Kingma, D.P. and Ba, J. (2014), "Adam: A method for stochastic optimization", *arXiv*: 1412.6980.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Conference on Neural Information Processing Systems*, NV, USA, December.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, **521**(7553), 436-444. https://doi.org/10.1038/nature14539

Lee, S.H., Lee, K.K., Woo, S.S. and Cho, S.H. (2013), "Global vertical mode vibrations due to human group rhythmic movement in a 39 story building structure", *Eng. Struct.*, **57**, 296-305. https://doi.org/10.1016/j.engstruct.2013.09.035

Li, G., Ji, T. and Chen, J. (2018), "Determination of the dynamic load factors for crowd jumping using motion capture technique", *Eng. Struct.*, **174**, 1-9. https://doi.org/10.1016/j.engstruct.2018.07.056

Mirza, M. and Osindero, S. (2014), "Conditional generative adversarial nets", *arXiv*: 1411.1784.

Parkhouse, J.G. and Ewins, D.J. (2006), "Crowd-induced rhythmic loading", *P. I. Civil Eng-Str. B.*, **159**(5), 247-259. https://doi.org/10.1680/stbu.2006.159.5.247

Press, O., Bar, A., Bogin, B., Berant, J. and Wolf, L. (2017), "Language generation with recurrent generative adversarial networks without pre-training", *arXiv*: 1706.01399.

Racic, V. and Pavic, A. (2010a), "Mathematical model to generate near-periodic human jumping force signals", *Mech. Syst. Signal Pr.*, **24**(1), 138-152. https://doi.org/10.1016/j.ymssp.2009.07.001

Racic, V. and Pavic, A. (2010b), "Stochastic approach to modelling of near-periodic jumping loads", *Mech. Syst. Signal Pr.*, **24**(8), 3037-3059. https://doi.org/10.1016/j.ymssp.2010.05.019

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), "Learning representations by back-propagating errors", *Nature*, **323**(6088), 533-536. https://doi.org/10.1038/323533a0

Sim, J., Blakeborough, A., Williams, M.S. and Parkhouse, G. (2008), "Statistical model of crowd jumping loads", *J. Struct. Eng.*, **134**(12), 1852-1861. https://doi.org/10.1061/(asce)0733-9445(2008)134:12(1852)

Van Nimmen, K., Lombaert, G., De Roeck, G. and Van den Broeck, P. (2014), "Vibration serviceability of footbridges: Evaluation of the current codes of practice", *Eng. Struct.*, **59**, 448-461. https://doi.org/10.1016/j.engstruct.2013.11.006

Wang, X., Pereira, E., Diaz, I.M. and Garcia-Palacios, J.H. (2018), "Velocity feedback for controlling vertical vibrations of pedestrian-bridge crossing. Practical guidelines", *Smart Struct. Syst.*, *Int. J.*, **22**(1), 95-103.

https://doi.org/10.12989/sss.2018.22.1.095
Wu, J., Zhang, C., Xue, T., Freeman, B. and Tenenbaum, J. (2016), "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling", *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, December.
Xiong, J. and Chen, J. (2018), "Power spectral density function for individual jumping load", *Int. J. Struct. Stab. Dy.*, **18**(2), 1850023. https://doi.org/10.1142/s0219455418500232
Xiong, J. and Chen, J. (2019), "Random field model for crowd jumping loads", *Struct. Saf.*, **76**, 197-209. https://doi.org/10.1016/j.strusafe.2018.10.001
Xiong, J. and Chen, J. (2021), "Open access and updated human-induced load data set", *J. Struct. Eng.*, **147**(3), 04720003. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002932
Xiong, J., Chen, J. and Caprani, C. (2021), "Spectral analysis of human-structure interaction during crowd jumping", *Appl. Math. Model.*, **89**, 610-626. https://doi.org/10.1016/j.apm.2020.07.030
Ye, X.W., Dong, C.Z. and Liu, T. (2016), "Image-based structural dynamic displacement measurement using different multi-object tracking algorithms", *Smart Struct. Syst.*, *Int. J.*, **17**(6), 935-956. https://doi.org/10.12989/sss.2016.17.6.935

Appendix A

Table A1 Probability mass functions of jumping frequency ($\times 10^{-4}$)

F_{jump} [Hz] \ Test [Hz]	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8
1.30	67	0	0	0	0	0	0	0	0	0	0	0	0	0
1.35	267	66	0	0	0	0	0	0	0	0	0	0	0	0
1.40	67	0	0	0	0	0	0	0	0	0	0	0	0	0
1.45	333	132	50	0	0	0	0	0	0	0	0	0	0	0
1.50	**4600**	66	50	0	0	0	0	0	0	0	0	0	0	0
1.55	1933	132	50	0	0	0	0	0	0	0	0	0	0	0
1.60	1067	**3907**	149	0	0	0	0	0	0	0	0	0	0	0
1.65	467	2053	299	61	0	0	0	0	0	0	0	0	0	0
1.70	267	1258	**5821**	183	91	0	0	0	0	0	0	0	0	0
1.75	67	265	995	0	46	0	0	0	0	0	0	0	0	0
1.80	133	530	995	**5854**	91	0	0	0	0	0	0	0	0	0
1.85	200	530	348	1220	183	48	118	0	0	0	0	0	0	0
1.90	0	397	299	915	**7489**	242	118	0	0	0	0	0	0	0
1.95	133	265	149	305	1096	290	0	0	0	0	0	0	0	0
2.00	67	66	249	488	320	**7198**	59	58	0	0	0	0	0	0
2.05	67	66	199	488	183	870	294	174	121	60	0	0	46	0
2.10	0	66	100	305	183	386	**8000**	0	182	60	0	90	0	0
2.15	0	0	0	122	183	145	647	523	121	120	0	0	0	0
2.20	0	0	50	0	0	242	59	**8314**	242	0	0	0	139	0
2.25	0	0	50	0	0	242	294	349	242	180	91	90	46	46
2.30	0	0	50	61	91	97	59	174	**8303**	120	364	0	0	0
2.35	0	0	0	0	0	0	176	58	121	180	273	45	93	46
2.40	0	0	0	0	0	48	59	116	121	**8443**	91	226	139	92
2.45	0	0	50	0	0	97	59	58	242	299	409	271	231	46
2.50	0	0	0	0	0	48	59	0	61	60	**8409**	90	93	229
2.55	0	0	0	0	46	48	0	116	121	60	182	543	231	92
2.60	67	0	0	0	0	0	0	0	61	60	0	**8054**	185	138
2.65	0	0	0	0	0	0	0	58	61	180	182	90	648	46
2.70	0	0	0	0	0	0	0	0	0	0	0	181	**7639**	92
2.75	133	0	0	0	0	0	0	0	0	60	0	90	139	596
2.80	0	0	0	0	0	0	0	0	0	60	0	90	46	**7844**
2.85	0	0	0	0	0	0	0	0	0	0	0	90	46	367
2.90	0	0	0	0	0	0	0	0	0	0	0	0	139	92
2.95	0	132	0	0	0	0	0	0	0	60	0	45	0	46
3.00	67	66	0	0	0	0	0	0	0	0	0	0	46	92
3.05	0	0	0	0	0	0	0	0	0	0	0	0	46	0
3.10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.15	0	0	0	0	0	0	0	0	0	0	0	0	46	46
3.20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.35	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.40	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.45	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.55	0	0	50	0	0	0	0	0	0	0	0	0	0	46
3.60	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.65	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.70	0	0	0	0	0	0	0	0	0	0	0	0	0	46