An Intelligent Ranking Evaluation Method of Simulation Models Based on Graph Neural Network

Fan Yang[1,2], Ping Ma[1,2], Jianchao Zhang[3], Huichuan Cheng[3], Wei Li[1,2(✉)], and Ming Yang[1,2]

[1] Control and Simulation Center, Harbin Institute of Technology, Harbin 150080, China
fleehit@163.com
[2] National Key Laboratory of Complex System Modeling and Simulation, Harbin 150080, China
[3] Chinese Aeroengine Research Institute, Beijing 101304, China

Abstract. To validate the alternative simulation models and select the most credible one when the models have multivariate and correlated outputs, an intelligent ranking evaluation method of simulation models based on Graph Neural Network (GNN) is proposed. The process of ranking evaluation is divided into three parts: graph structure conversion for evaluation data, feature extraction based on Graph Representation Learning (GRL) and ranking evaluation based on feature distance. A graph structure modeling method is presented to provide the predefine graph structure for further GRL primarily. Next the interdependencies and dynamic evolutionary patterns among variables are captured by GNN so that the graph representations of evaluation data can be obtained. Then ranking evaluation is achieved by similarity measurement of the graph representations. In the end, the effectiveness of the proposed method on feature extraction of evaluation data and simulation models ranking is illustrated through an application example on a prediction model for aerodynamic parameters of a certain flight vehicle.

Keywords: Ranking Evaluation of Simulation Models · Multivariate and Correlated Outputs · Graph Neural Network (GNN)

1 Introduction

Simulation is an important means of assisting complex systems in design, development, analysis, evaluation, optimization, and decision-making [1] and has been widely used in various fields such as military, manufacturing, medical, transportation, etc. In practical applications, the uncertainty of model parameters and the diversity of modeling methods may lead to the ranking and selection problem of alternative simulation models. It is necessary to validate the credibility of multiple alternative simulation models and select the most credible one [2].

Simulation result validation method is generally used for the simulation model validation. The principle of the method is to obtain the credibility of the simulation model

J.-L. Guisado-Lizar et al. (Eds.): SIMUtools 2023, LNICST 519, pp. 120–130, 2024.
https://doi.org/10.1007/978-3-031-57523-5_10

by measuring the consistency between simulation outputs and reference outputs. The evaluation data of continuous system simulation such as aircraft motion simulation and guidance control system simulation are mostly in the form of time series with correlation [3]. However, the existing multivariate validation methods generally focus on the validation of multivariate static data [4, 5], such as multivariate Bayesian hypothesis testing [6], probability integral transformation (PIT) and area metric [7], principal component analysis and area metric [8], etc.

The validation of multivariate simulation results is essentially a similarity measurement problem for multivariate time series (MTS). Some common methods achieve multivariate result validation by synthesizing the validation result of univariate [9] while ignoring the correlation between multiple variables. Some other methods analyze the similarity after reducing the dimension of multivariate outputs [3, 10] which may lead to loss of information, and some of these methods may be computationally inefficiency for large kernel matrix [3]. There is also a type of simulation result validation method based on data feature [11]. However, it is difficult to select suitable features for evaluation data artificially. There is still a lack of effective validation methods for multivariate dynamic outputs with correlation.

Deep learning can autonomously and quickly learn effective features in multi-level abstraction process. Recently, Graph Neural Network (GNN) has received increasing attention in modeling MTS to achieve MTS prediction [12, 13] and anomaly detection [14, 15] due to high capability in dealing with relational dependencies [16]. How to use GNN to extract the features of the evaluation data from different models could be considered as a supervised and graph-level Graph Representation Learning (GRL) task to be study. In addition, GNNs are generally performed on data with a pre-defined graph structure while the relationships among output variables are generally unknown. How to convert evaluation data into graph-structured data is also a problem that needs to be solved.

In summary, we expect to achieve ranking evaluation of multiple alternative simulation models under correlation based on data feature and extract the features of evaluation data by GNN. Firstly, we propose a graph structure modeling method based on distance correlation coefficient to provide the pre-define graph structure for further GRL. Then, we adopt GRL method based on Graph Isomorphism Network (GIN) [17] to extract the features of evaluation data. Next, the alternative simulation models could be ranked by comparing the similarity of the features corresponding to different simulation models and the real system.

The rest of the paper is organized as follows. The ranking evaluation method based on GNN is presented in Sect. 2, including the problem description in Sect. 2.1, the graph structure modeling method in Sect. 2.2, the feature extraction method in Sect. 2.3 and the ranking evaluation method in Sect. 2.4. Then, an application example is enumerated to verify the effectiveness of the proposed method in Sect. 3. Finally, the conclusion and the future work are given in Sect. 4.

2 Ranking Evaluation Method Based on GNN

2.1 Problem Description

Suppose that S denotes the system, and $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\right]^{\mathrm{T}}$ represents M outputs of S. $S_s = \{S_{s1}, S_{s2}, \ldots S_{sA}\}$ and S_r are regarded as the alternative simulation models and the real system. Let the real system and all alternative simulation models run repeatedly under several given inputs, $\mathbf{Y}_{sa} = \left[\mathbf{y}_{sa1}, \mathbf{y}_{sa2}, \ldots, \mathbf{y}_{saM}\right]^{\mathrm{T}}$ and $\mathbf{Y}_r = \left[\mathbf{y}_{r1}, \mathbf{y}_{r2}, \ldots, \mathbf{y}_{rM}\right]^{\mathrm{T}}$ represent the M outputs of $S_{sa}(a = 1, 2, \ldots A)$ and S_r respectively while only dynamic outputs are considered in this problem. $\mathbf{Y}_{sa}$ is a $T \times M \times N_s$ matrix representing M simulation output variables across T timestamps each having N_s samples for aleatory uncertainty. Similarly, $\mathbf{Y}_r$ is a $T \times M \times N_r$ matrix representing M reference output variables across T timestamps each having N_r samples for aleatory uncertainty. Suppose that $C(\mathbf{Y}_{sa}, \mathbf{Y}_r)$ denotes the credibility of the simulation model S_{sa} which is measured by the consistency between $\mathbf{Y}_{sa}$ and $\mathbf{Y}_r$. S_{sa} is the most credible one when $C(\mathbf{Y}_{sa}, \mathbf{Y}_r)$ is largest in $\{C(\mathbf{Y}_{s1}, \mathbf{Y}_r), C(\mathbf{Y}_{s2}, \mathbf{Y}_r), \ldots C(\mathbf{Y}_{sA}, \mathbf{Y}_r)\}$.

To solve the ranking evaluation problem of multiple alternative simulation models, we adopt GNN to extract the features of evaluation data and obtain ranking evaluation result by comparing feature distance. The outputs of each alternative simulation model can be represented by some graphs while reference outputs can be represented by other graphs. Given two sets of graphs $\mathbf{G}_{sa} = \left\{\mathbf{G}_{sa}^1, \mathbf{G}_{sa}^2, \ldots \mathbf{G}_{sa}^{N_s}\right\}$, $\mathbf{G}_r = \left\{\mathbf{G}_r^1, \mathbf{G}_r^2, \ldots \mathbf{G}_r^{N_r}\right\}$ and assuming that $\mathbf{h}_{Gsa}$ and $\mathbf{h}_{Gr}$ represent the features of two sets of outputs respectively, $D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr})$ represents the corresponding feature distance. It is certain that the smaller $D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr})$ is, the larger $C(\mathbf{Y}_{sa}, \mathbf{Y}_r)$ is.

An overview of ranking evaluation framework is shown in Fig. 1. The process of ranking evaluation mainly includes three stages, data convertion, feature extraction and similarity comparison. We convert the evaluation data into graph structured data and then learn the graph representations which can effectively represent the features of the evaluation data. Furthermore, ranking evaluation of the alternative simulation models could be achieved by comparing the similarity of graph representations corresponding to different simulation models and the real system.

2.2 Graph Structure Modeling for Evaluation Data

The evaluation data need to be converted from two-dimensional form to graph structure for further GRL. Graph is a non-linear data structure which has proved useful for analyzing multivariate and correlated data. Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ denotes a graph which consists of a finite non empty set of nodes $v_i \in \mathbf{V}$ and edges $e_{ij} \in \mathbf{E}$. And $e_{ij} = (v_i, v_j)$, which means that e_{ij} denotes the edge between v_i and v_j. In order to analyze the evaluation data, we proposed a graph structure modeling method in which output variables and their correlation are represented by nodes and edges respectively.

Given $\mathbf{y}^n$, a sample of the evaluation data, the component $\mathbf{y}_i^n (i \in [1, M])$ is a $1 \times T$ matrix which represents the i_{th} output variable across T timestamps. In the corresponding graph $\mathbf{G}^n$, $\mathbf{y}_i^n$ is represented by a node $v_i \in \mathbf{V}$. The number of nodes in the graph corresponds to the number of output variables and thus $\mathbf{V} = \{v_1, v_2, \ldots, v_M\}$.

$$\mathbf{y}_i^n = \left[y_i^n(1)\, y_i^n(2) \ldots y_i^n(T)\right] \tag{1}$$

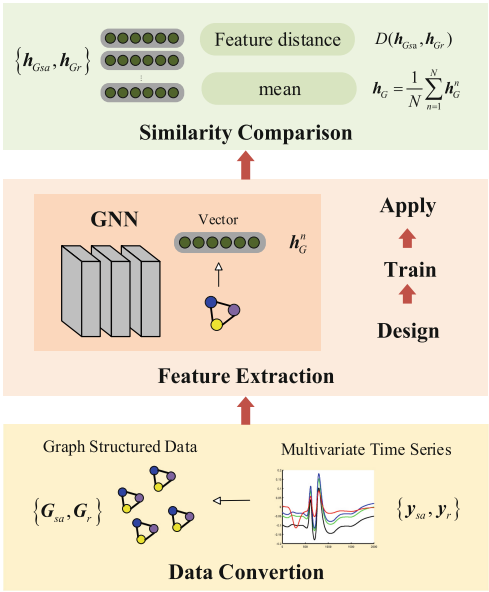


Fig. 1. An overview of ranking evaluation framework.

The graph structure can describe the relationships among multiple variables. There exists an edge $e_{ij}(i, j \in [1, M])$ between node v_i and v_j when the correlation between $\mathbf{y}_i^n$ and $\mathbf{y}_j^n$ exceeds a certain threshold. We adopt the distance correlation coefficient [18] to measure the correlation. The larger the distance correlation coefficient, the higher the correlation between the two variables. Besides, the threshold could be set differently in different problems.

The adjacency matrix, $\mathbf{A} = \{A_{ij}\} \in \mathbb{R}^{M \times M}$, is a storage structure that uses a matrix to represent the relationships among the nodes in a graph and the local-level representations are learned based on the adjacent relationships. Each element A_{ij} represents the relationship between node v_i and v_j. If $A_{ij} = 1$, there exists an edge between v_i and v_j, and $A_{ij} = 0$ otherwise.

We set the value of A_{ij} based on the distance correlation coefficient of the corresponding paired nodes and the given threshold m_d. Given the outputs of the i_{th} variable and the j_{th} variable in a sample of outputs $\mathbf{y}_{sa}^n$ or $\mathbf{y}_r^n$, we can obtain the distance correlation coefficient between node v_i and v_j in the corresponding graph. The distance correlation coefficient is calculated as follow:

$$dCor(\mathbf{y}_i, \mathbf{y}_j) = \frac{dCov(\mathbf{y}_i, \mathbf{y}_j)}{\sqrt{dCov(\mathbf{y}_i, \mathbf{y}_i) dCov(\mathbf{y}_j, \mathbf{y}_j)}} \tag{2}$$

where $dCor(\mathbf{y}_i, \mathbf{y}_j) \in [0, 1]$, $\mathbf{y}_i = \{y_i(p)\}$ and $\mathbf{y}_j = \{y_j(q)\}$ represent two variables, and $dCov^2(\mathbf{y}_i, \mathbf{y}_j) = S_1 + S_2 - 2S_3$ with S_1, S_2, S_3 defined as follows:

$$S_1 = \frac{1}{T^2} \sum_{p=1}^{T} \sum_{q=1}^{T} |y_i(p) - y_i(q)| \left| y_j(p) - y_j(q) \right| \tag{3}$$

$$S_2 = \frac{1}{T^2}\sum_{p=1}^{T}\sum_{q=1}^{T}|y_i(p) - y_i(q)|\frac{1}{T^2}\sum_{p=1}^{T}\sum_{q=1}^{T}\left|y_j(p) - y_j(q)\right| \tag{4}$$

$$S_3 = \frac{1}{T^3}\sum_{p=1}^{T}\sum_{q=1}^{T}\sum_{o=1}^{T}|y_i(p) - y_i(o)|\left|y_j(p) - y_j(o)\right| \tag{5}$$

Furthermore, let $A_{ij} = 1$ when $dCor > m_d$, and $A_{ij} = 0$ otherwise.

2.3 Feature Extraction Based on GRL Model

Design of GRL Model Based on GIN. Each set of evaluation data corresponding a simulation model or real system can be represented by a set of graphs based on the graph structure modeling method. Given a set of graphs, and a positive integer D (the expected graph embedding size), our goal is to design a GRL model to learn a D-dimensional distributed graph representation.

To avoid the impact of different variable magnitudes on the results of feature extraction, normalize each variable as follow:

$$\mathbf{y}_i = \frac{\mathbf{y}_i - y_{\min}}{y_{\max} - y_{\min}} \tag{6}$$

where $\mathbf{y}_i$ is a sample of evaluation data corresponding to node v_i, $y_{\max}$ and $y_{\min}$ denote the maximum and minimum element in $\mathbf{y}_i$ respectively.

GIN is adopted to extract local features for its higher graph-level representational power. In GIN, the representation of nodes is updated by concatenating their own features with the features of neighbor nodes [19] and inputting them into an MLP for nonlinear transformation. Then the entire graph embedding could be calculated by readout function based on the representation of nodes.

We design the GIN layers according to output variables. When the number of output variables is large, the layers of GIN can be increased to obtain more information in larger scale subgraph. We input the preprocessed data into the GIN layers as Eq. (7), then the k_{th} GIN layer $g^{(k)}$ updates the representation of node v_i as Eq. (8):

$$\mathbf{h}_i^{(0)} = \mathbf{y}_i \tag{7}$$

$$\begin{aligned}\mathbf{h}_i^{(k)} =& g^{(k)}(\mathbf{h}_i^{(k-1)}, \mathbf{h}_j^{(k-1)} | j \in \mathbf{N}_i) \\ =& MLP^{(k)}((1+\varepsilon^{(k)}) \cdot \mathbf{h}_i^{(k-1)} + \sum\nolimits_{j\in N_i} \mathbf{h}_j^{(k-1)})\end{aligned} \tag{8}$$

where $MLP^{(k)}$ represents the MLP in the k_{th} GIN layer $g^{(k)}$, $\boldsymbol{\varepsilon}^{(k)}$ is trainable aggregation parameter in the k_{th} GIN layer $g^{(k)}$, $\mathbf{h}_i^{(k-1)}$ is the output of the $(k-1)_{th}$ GIN layer $g^{(k-1)}$, and $\mathbf{N}_i$ is a set of nodes adjacent to v_i.

After applying L GIN layers, we can consider the output of the last layer $\mathbf{h}_i^{(L)}(i = 1, 2, ...M)$ as the updated node representations which represent the local features. This process can be described as Eq. (9).

$$g(\mathbf{h}_i^{(0)}) = \mathbf{h}_i^{(L)}(i = 1, 2, ...M) \tag{9}$$

It is necessary to summarize these patch representations into the representation of the entire graph by a readout function so that we can obtain a fixed length feature vector as follow for further similarity measurement.

$$\mathbf{h}_G = readout(\mathbf{h}_i^{(L)} | i = 1, 2, ...M) \tag{10}$$

In the majority of GRL models, $readout(\cdot)$ can be a simple function such as sum, mean, max, etc. or a more sophisticated graph-level pooling function [20, 21]. In this work, we use sum as the readout function to catch global information. Overall, we have designed the structure of GNN for feature extraction as Fig. 2.

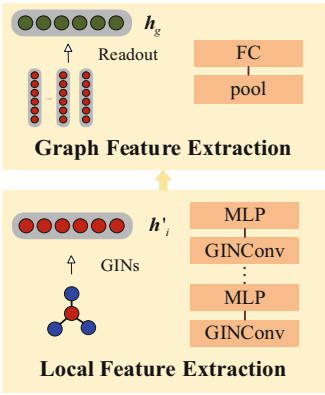


Fig. 2. Structure of GRL model.

Training and Application of GRL Model. We hope that the trained GRL model can effectively extract the features of the evaluation data and the features of the evaluation data from the same simulation model are similar while the features of the evaluation data from different simulation models are dissimilar. Then we could apply the trained GRL model to extract the features of the simulation outputs and the reference outputs and compare the consistency of two sets of features. Hence the ranking evaluation problem of multiple alternative simulation models is transformed into the evaluation data multi-categorization problem and the class of labels are equal to the number of simulation models. There may be a variety of input conditions for simulation experiments, however we assign the evaluation data generated from the same simulation model to the same labels regardless of the input conditions.

In the training stage, we input the graph-structured simulation outputs into the designed GRL model to obtain graph representation. In order to ensure that the graph representation can effectively represent the features of the evaluation data to support the subsequent similarity measurement task, the cross-entropy loss is used as the objective function as Eq. (11) and the stochastic gradient descent algorithm is used to train the

GRL model as shown in Fig. 3.

$$L = \frac{1}{N} \sum_{n} \sum_{a=1}^{A} y_{na} \log(p_{na}) \tag{11}$$

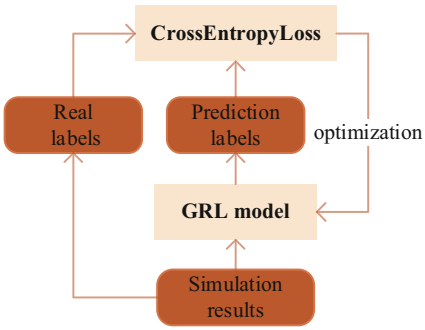


Fig. 3. Training and optimization for GRL models.

Where A is the number of categories, y_{na} denotes the symbolic function as Eq. (12), and p_{na} denotes the predicted probability that the sample y_n belongs to category a as Eq. (13).

$$y_{na} = \begin{cases} 0 \; y_n \text{ is not generated from model } S_{sa} \\ 1 \; y_n \text{ is generated from model } S_{sa} \end{cases} \tag{12}$$

$$p_{na} = -\log\left(\frac{e^{x_a}}{\sum_{i=0}^{A} e^{x_i}}\right) = -x_a + \log\left(\sum\nolimits_{i=0}^{A} e^{x_i}\right) \tag{13}$$

Then the trained model can be applied to extract the features of evaluation data. In the application stage, we input two sets of graph-structured data which represent simulation outputs and reference outputs into the trained GRL model to obtain two sets of graph representations, $\mathbf{h}^n_{Gsa}(n = 1, 2, ...N_s, a = 1, 2, ..., A)$ and $\mathbf{h}^n_{Gr}(n = 1, 2, ...N_r)$.

2.4 Ranking Evaluation Based on Feature Distance

Since the two sets of graph representations $\mathbf{h}^n_{Gsa}(n = 1, 2, ...N_s)$ and $\mathbf{h}^n_{Gr}(n = 1, 2, ...N_r)$ can effectively represent the features of the outputs from the simulation model S_{sa} and reference outputs, the validation result of S_{sa} can be obtained by comparing the similarity of two sets of graph representations. To obtain the overall features of the simulation outputs from S_{sa} and the reference outputs, we calculate the mean of two sets of graph representations due to multi samples as follows:

$$\mathbf{h}_{Gsa} = \frac{1}{N_s} \sum_{n=1}^{N_s} \mathbf{h}^n_{Gsa} \tag{14}$$

$$\mathbf{h}_{Gr} = \frac{1}{N_r} \sum_{n=1}^{N_r} \mathbf{h}_{Gr}^{n} \tag{15}$$

Then 2-norm distance is adopted to measure the feature distance $D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr})$ as Eq. (16). The distance between $\mathbf{h}_{Gsa}^{n}$ and $\mathbf{h}_{Gr}^{n}$ represents the consistency between the corresponding simulation outputs $\mathbf{Y}_{sa}$ and reference outputs $\mathbf{Y}_{r}$ so that $D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr})$ can be used to describe $C(\mathbf{Y}_{sa}, \mathbf{Y}_{r})$. And the smaller $D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr})$ is, the larger $C(\mathbf{Y}_{sa}, \mathbf{Y}_{r})$ is.

$$D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr}) = \|\mathbf{h}_{Gsa} - \mathbf{h}_{Gr}\|_2 \tag{16}$$

Furthermore, we can rank multiple alternative simulation models by comparing the value of $\{D(\mathbf{h}_{Gs1}, \mathbf{h}_{Gr}), D(\mathbf{h}_{Gs2}, \mathbf{h}_{Gr}), ..., D(\mathbf{h}_{GsA}, \mathbf{h}_{Gr})\}$. The smaller $D(\mathbf{h}_{Gsa}, \mathbf{h}_{Gr})$ is, the more credible the simulation model S_{sa} is.

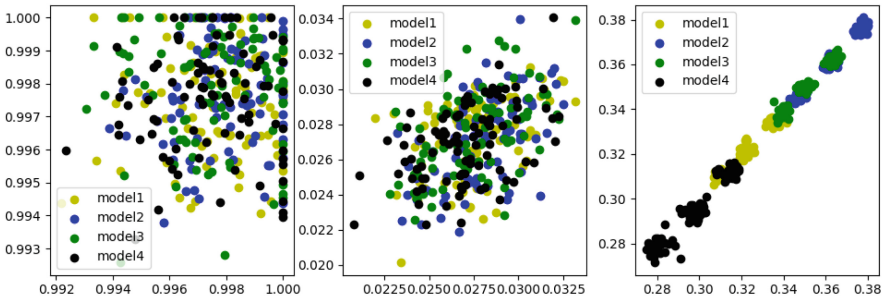
3 Experimental Studies

In this section, we evaluate the effectiveness of the proposed ranking evaluation method by an application example on the aircraft aerodynamic parameter prediction model. There are four simulation models in the experiment including an identification model and three prediction models, and the prediction models include the 5-s prediction model, the 8-s prediction model, and the 10-s prediction model. The prediction models are used to predict the outputs at the current moment according to the historical information a few seconds ago. Given the labels of these four models are model 1, model 2, model 3, model 4.

Compared to the prediction models, the identification models are significantly more credible. In addition, among three prediction models, the longer the prediction time, the less credible the prediction model is. It is obvious that the ranking result of the four models should be in the order of model 1, model 2, model 3 and model 4 so that we apply the proposed method to evaluate these simulation models to validate the effectiveness of the method.

We select three outputs for simulation ranking evaluation including lift coefficients, drag coefficients and moment coefficients. Given 3 input conditions, let these four models run 10 times under each condition respectively while the real system run 5 times under each condition so that 135 sets of outputs in the range of 10 s to 90 s with a sampling interval of 0.02 s were obtained. Each set of evaluation data is a 4001×3 matrix representing 3 output variables across 4001 timestamps. The first two dimensions of each variable in the four simulation models are taken for observation before training, and the data distribution is shown in Fig. 4.

We analyze the correlation between paired variables by reference data based on Eq. (2)–Eq. (5), and the mean of the paired distance correlation coefficients among the three variables are 0.971, 0.989, 0.991 respectively. The value of each element in the adjacency matrix could be set as 1 with $m_d = 0.9$. There are significant correlations among the three variables and it is in good agreement with practice. After that we have completed the data convertion.



(a) Value of lift coefficients (b) Value of drag coefficients (c) Value of moment coefficients

Fig. 4. The distribution of original outputs.

Two GIN layers are adopted for feature extraction and the simulation outputs are used for training which is a 4001 × 3 × 120 matrix. The parameters of each layer are set as Table 1 after training and optimization. Training terminates when the number of iterations reaches 400.

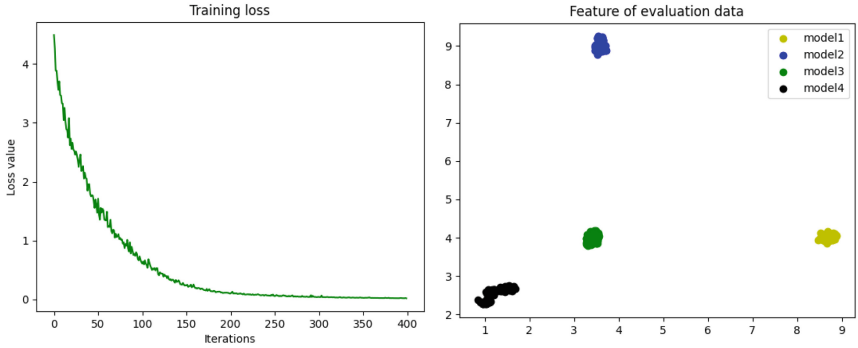
Table 1. The parameters of GNN.

layer	Linear1	Linear2
ginconv_1	[4001,1024]	[1024,256]
ginconv_2	[256,64]	[64,64]

Then we can get the change of the loss value during the training process and the distribution of the first two dimensions of the evaluation data feature at the end of training (see Fig. 5). The feature distributions of the outputs of the same model under different conditions are close while the feature distribution of the outputs of different models are farther away. It shows that the GRL method can effectively extract the features of multivariate correlated variables.

Inputting the reference data which is a 4001 × 3 × 15 matrix into the trained GRL model, then the features of the reference data can be obtained. Furthermore, the feature distance could be calculated based on Eq. (14–16). The features corresponding to the outputs under the three conditions are taken to calculate the feature distance respectively, and the results are shown in Table 2.

It means that the sorting result of the above simulation models is model 1, model 2, model 3 and model 4 which is consistent with the fact. It can be seen that the ranking evaluation method of simulation models based on GNN can effectively achieve the ranking evaluation under the multivariate and correlated outputs.



(a) Loss during training (b) The distribution of features

Fig. 5. Results of training based on GNN.

Table 2. Feature distance under different conditions.

	Model1	Model2	Model3	Model4
Condition1	0.15	2.33	3.82	6.33
Condition2	0.63	2.56	3.90	6.50
Condition3	1.70	2.46	3.97	6.17
Mean	0.83	2.45	3.90	6.33

4 Conclusions

To solve the ranking and selection problem of multiple alternative simulation models with multivariate and correlated outputs, an intelligent ranking evaluation method of simulation models based on GNN has been presented. We propose a ranking evaluation framework to illustrate the process of ranking evaluation for multiple alternative simulation models in which there are three stages including data convertion, feature extraction and similarity comparison.

The proposed method is applied to ranking four models about the aircraft aerodynamic parameter prediction. Obviously, the method is effective to extract the features of the evaluation data and rank multiple alternative simulation models with multivariate and correlated outputs. The future work would focus on solving the ranking evaluation of simulation models with the multivariate, heterogeneous and correlated outputs.

Funding Information. National Science and Technology Major Project(J2019-I-0004-0005).

References

1. Sargent, R.G.: Verification and validation of simulation models. J. Simul. **7**(1), 12–24 (2013)
2. Fan, Y., Ping, M., Wei, L., et al.: An intelligent ranking evaluation method for simulation models based on twin networks. Syst. Eng. Electron. **45**(07), 2060–2068 (2023)

3. Yuchen, Z.: Research on Verification Methods for Complex Simulation Models. Harbin Institute of Industry, Harbin (2019)
4. Rebba, R., Mahadevan, S.: Validation of models with multivariate output. Reliab. Eng. Syst. Saf. **91**(8), 861–871 (2006)
5. Oberkampf, W.L., Barone, M.F.: Measures of agreement between computation and experiment: validation metrics. J. Comput. Phys. **217**(1), 5–36 (2006)
6. Jiang, X.M., Mahadevan, S.: Bayesian wavelet method for multivariate model assessment of dynamic systems. J. Sound Vib. **312**(4–5), 694–712 (2008)
7. Wei, L., Zhenzhou, L., Wei, C.: New validation metrics for models with multiple correlated responses. Reliab. Eng. Syst. Saf. **127**, 1–11 (2014)
8. Luyi, L., Zhenzhou, L.: A new method for model validation with multivariate output. Reliab. Eng. Syst. Saf. **169**, 579–592 (2018)
9. Kai, Z., Jie, H., Zhenfei, Z.: Multivariate response analysis for dynamic system model validation. J. Shanghai Jiao Tong Univ. **02**, 191–195 (2015)
10. Haiying, W.: Research on verification method of multi variable uncertainty simulation results. Harbin Institute of Technology, Harbin (2016)
11. Shenglin, L., Wei, L., Ming, Y., Ping, M.: Validation method for multivariate output simulation models considering correlation. J. Autom. **45**(09), 1666–1678 (2019)
12. Xiulin, G., Xiaoyu, H., Lingyu, X.: Graph correlated attention recurrent neural network for multivariate time series forecasting. Inf. Sci. **606**, 126–142 (2022)
13. Zonghan, W., Shirui, P., Guodong, L.: Connecting the dots: multivariate time series forecasting with graph neural networks. In: The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 753–763 (2020)
14. Liwei, Z., Qingkui, Z., Bo, L.: Hybrid anomaly detection via multihead dynamic graph attention networks for multivariate time series. IEEE Access **10**, 40967–40978 (2022)
15. Jun, Z., Siqi, W., Xiandong, M., et al.: Stgat-Mad: spatial-temporal graph attention network for multivariate time series anomaly detection. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3568–3572. Singapore, Singapore (2022)
16. Mikalsen, K.Y., Bianchi, F.M., Soguero, R.C.: Time series cluster kernel for learning similarities between multivariate time series with missing data. Pattern Recognit. **76**, 569–581 (2018)
17. Xu, K., Hu, W., Leskovec, J., et al.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019)
18. Szekely, G.J., Rizzo, M.L., Bakirov, N.K.: Measuring and testing dependence by correlation of distance. Ann. Stat. **35**(6), 2769–2794 (2008)
19. Gilmer, J., Schoenholz, S.S., Riley, P.F.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning, pp. 2053–2070 (2017)
20. Zhitao, Y., Jiaxuan, Y., Christopher, M., et al.: Hierarchical graph representation learning with differentiable pooling. In Advances in Neural Information Processing Systems, pp. 4800–4810 (2018)
21. Muhan, Z., Zhicheng, C., Marion, N., et al.: An end-to-end deep learning architecture for graph classification. In: In Thirty-Second AAAI Conference on Artificial Intelligence (2018)