Data-driven hysteretic behavior simulation based on weighted stacked pyramid neural network architecture

Yongjia Xu 1, Xinzheng Lu 2*, Yifan Fei 1, Yuli Huang 2

1. Beijing Engineering Research Center of Steel and Concrete Composite Structures, Tsinghua University, Beijing, P.R. China, 100084
2. Department of Civil Engineering, Tsinghua University, Beijing, P.R. China, 100084

Abstract

An accurate and efficient simulation of the hysteretic behavior of materials and components is essential for structural analysis. The surrogate model based on neural networks shows significant potential in balancing efficiency and accuracy. However, its serial information flow and prediction based on single-level features adversely affect the network performance. Therefore, a weighted stacked pyramid neural network architecture is proposed herein. This network establishes a pyramid architecture by introducing multi-level shortcuts to directly integrate features in the output module. In addition, a weighted stacked strategy is proposed to replace the conventional feature fusion method. The weights of the features are determined based on their levels. These basic principles are verified, and key network settings are discussed. Subsequently, the redesigned architectures are compared with other commonly used algorithms. Results show that the testing mean-square error (MSE) loss of the networks on varied datasets can be reduced by an average of 34.7%. The redesigned architectures outperform 87.5% of cases, and the proposed Pyramid-GA network has the best overall performance.

Keywords: Hysteretic behavior simulation; Multi-level shortcut; Weighted-stacked feature fusion; Pyramid neural network architecture

Corresponding author: Xinzheng Lu, Room No.415, Heshanheng building, Department of civil engineering, Tsinghua university, Haidian district, Beijing, P.R. China, 100084, E-mail: luxz@tsinghua.edu.cn
1. Introduction

An accurate and efficient simulation of hysteretic behavior is essential for structural (or component/material) analysis. Significant progress has been realized in refined physical models (e.g., Tao and Shahsavari, 2017; Li, 2018). Meanwhile, the complexity of refined models has increased considerably, resulting in an obvious increase in the computational workload.

To balance the accuracy and efficiency, many studies have focused on data-driven surrogate models. Conventional surrogate models are generally developed based on experimental data and simulation results (e.g., Long and Lee, 2012; Pham et al., 2016; Liu and Guo, 2021; Lu and Guan, 2021), which exhibits certain disadvantages in terms of universality and accuracy: (1) the model simplification process relies significantly on artificially determined function forms and key assumptions or parameters; (2) for hysteretic behaviors with complex features and limited data, a simplified and accurate surrogate model is difficult to be obtained.

Deep learning has developed rapidly in recent years. The deep neural network model involves numerous parameters, exhibits strong nonlinear fitting ability and “end-to-end” characteristics (from the input to output directly, and delivers a complete solution without artificially managing the intermediate process), thereby effectively overcoming the abovementioned disadvantages of conventional surrogate models. Therefore, deep learning methods have garnered increasing attention.

Similar to natural language processing, the essence of hysteretic behavior simulation is “sequence modeling”. Therefore, the mainstream neural networks adopted in similar studies are recurrent neural network (RNN) (particularly the long short-term memory (LSTM) neural network (Hochreiter and Schmidhuber, 1997), gated recurrent unit (GRU) network (Cho et al., 2014)), and Transformer (or modified Transformer) (e.g., Vaswani et al., 2017; Wang and Sun, 2018; Zhang et al., 2019; Gorji et al., 2020; Xu et al., 2021). Other technologies, such as the multi-layer perception (MLP) and convolutional neural network (CNN), are also reported (e.g., Zhang et al., 2020; Xu et al., 2020; Lu et al., 2021).

In these aforementioned networks, the forward propagation is primarily serial (i.e., from the encoder to the decoder, and from the first layer to the final layer), and the prediction is
conducted based on single-level features. Relevant studies in the field of computer vision (e.g., Ronneberger et al., 2015; He et al., 2016; Tan et al., 2020) show that multi-level feature fusion improves the prediction performance, and “Shortcut” is a reliable multi-level feature extraction and fusion technology. A neural network that integrates multi-level features can be termed as a “pyramid network”.

Shortcuts have been added to LSTM or Transformer networks for time-series modeling in some existing studies. For example, the classic Transformer architecture (Vaswani et al., 2017) involves intra-module shortcuts. Kim et al. (2017) and Zhao et al. (2018) introduce shortcuts in LSTM to improve the network performance. Emelin et al. (2019) propose lexical shortcut mechanism. Hao et al. (2019) propose a new module known as the recurrence encoder and transfer its outputs to the attention layer in the decoder through shortcuts. Liu et al. (2020) integrate the outputs of LSTM and the attention module for prediction through shortcuts.

However, existing studies exhibit the following limitations:

1) In sequence modeling, shortcuts are primarily added within certain modules, whereas shortcuts that forego multiple layers/modules or connect the encoder and decoder are relatively rare (such multi-level shortcuts mostly exist in the field of computer vision).

2) Shortcuts are primarily used to transfer information across intermediate layers/modules, whereas intermediate features are rarely used for output prediction directly. In other words, the output module can only fulfill prediction based on the output feature of the final module.

3) The feature fusion mechanism considers different levels of features equally, which disregards the variety among different levels of features.

Therefore, this study aims to overcome these shortcomings to further improve the reliability of hysteretic behavior simulations based on deep learning. A weighted stacked pyramid network architecture is proposed herein. This architecture introduces multi-level shortcuts that directly connect the intermediate layers and the output module. In addition, a weighted stacked strategy is adopted to replace the conventional feature fusion method.

2. Methodology
2.1 Supporting Technology

LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014) are two typical improved networks of classic RNN, which are widely used in sequence modeling. Meanwhile, inspired by the human attention mechanism, Bahdanau et al. (2015) propose the attention mechanism for the neural network to reasonably allocate network resources to key information. Vaswani et al. (2017) modify the attention mechanism and propose the Transformer network, which shows high learning ability in sequence modeling.

2.2 Multi-level shortcut

As mentioned above, the prediction of the LSTM and Transformer depends only on the features extracted by the final layer/module. After extraction and condensation, the feature of the final layer is the most closely associated with the output. However, because the learning ability of the network is not infinite, such features could not retain all key information (Fig. 1(a)). Therefore, an architecture with multi-level shortcuts is proposed herein. Based on conventional serial information propagation, shortcuts that directly connect the intermediate layer and output modules are added (Fig. 1(b)).

Integrating multi-level features allows the output module to realize final predictions based on more valuable information, thereby improving the network performance. The overall architecture of the redesigned pyramid networks with multi-level shortcuts is shown in Fig. 2(a). The concept of adding multi-level shortcuts is versatile, and the overall architecture could be
maintained under different situations. Only the basic unit in the networks need to be modified.

If the new architecture is redesigned based on a traditional Transformer (named “Pyramid-Transformer”), the basic unit in the network is shown in Fig. 2(b). Under certain situations, the simpler network LSTM could also be used as the fundamental network, and the redesigned architecture is named “Pyramid-LSTM”. The unit in Pyramid-LSTM is the same as the LSTM cell (Hochreiter and Schmidhuber, 1997). Furthermore, some existing studies have combined the LSTM/GRU and attention mechanism for hysteretic behavior simulation (e.g., Wang et al., 2020; Li et al., 2021a, 2021b). The basic unit in the GRU + attention (GA) network is similar to that in Pyramid-Transformer, while a GRU layer is added in each unit, as shown in Fig. 2(c). The redesigned network is named “Pyramid-GA”. It can be seen that multi-level features are directly fused in the output module for prediction in both networks.

(a) The proposed multi-level shortcut
(b) Unit in Transformer
It is noteworthy that the modules in Fig. 2(a) can be further partitioned, and the feature output of the submodules can be imported into the output module by adding shortcuts. In the following section of this study, the key settings (i.e., the number of layers and the method of adding shortcuts) are compared and analyzed.

2.3 Weighted stacked feature fusion mechanism

Traditional serial network only takes the last level of feature into consideration when predicting the output (as shown in Figs. 1(a) and 2(d)). In addition, in previous studies, feature fusion is generally performed by (1) element-wise summarization, (2) concatenating certain dimensions, and (3) decreasing/increasing the spatial size together with element-wise summarization (when the spatial dimensions of the features are inconsistent). As illustrated in Fig. 2(d), these classic mechanisms consider different levels of features equally.

Although the aforementioned fusion mechanisms are simple and straightforward, they disregard the differences in features at different levels. In a neural network, all layers are designed to extract and condense features from their inputs. Therefore, in many reasonably designed neural networks, the deeper the layer to which the features belong, the more significant is the inherent relationship between the outputs and features, and vice versa. It is noteworthy that although the features belonging to the shallow layers are important, they are...
primarily used as the input of the next layers. In other words, their positions are different.

Based on the analysis above, the different treatments of the features from different levels conform to the principle of neural network design. After introducing shortcuts that forego multiple layers/modules, the levels of the features exhibit larger variance. Therefore, a weighted feature fusion mechanism is proposed. This mechanism determines the weights of the features based on their levels, as shown in Fig. 2(d) and Equation (1). In terms of weight selection, referring to the typical principles of spatial size decay and channel number improvement in convolutional neural networks (e.g., Krizhevsky et al., 2012; He et al., 2016; Redmon and Farhadi, 2018), an exponential weight decay pattern, as expressed in Equation (2), is selected.

\[
f_{\text{inte}} = \frac{\sum_{i=1}^{n} f_i w_i}{\sum_{i=1}^{n} w_i}
\]

\[
w_i = \frac{1}{p^k} \quad (p \geq 1.0)
\]

where \( f_{\text{inte}} \) represents the integrated feature; \( f_i \) represents the \( i \)-th features used for feature fusion, such as the Encoder Outputs L1-L2 and Decoder Outputs L1-L4 shown in Fig. 2(a); \( n \) represents the total number of features; \( w_i \) represents the weight corresponding to \( f_i \); \( p \) is the weight decay factor, which will be discussed below; \( k \) is the number of modules/layers between the layer (module) corresponding to \( f_i \) and the output layer.

3. Dataset Establishment

All datasets used in this study are summarized in Table 1 (they can been downloaded through https://github.com/XYJ0904/Weighted-Pyramid-Stacked-Network). Typical samples in these datasets are visualized in Section 5.
Table 1 Summarization of the basic information of all datasets

| Case Name | Scale | Input/Output | Series Length | Dataset size | References |
|-----------|-------|--------------|---------------|--------------|------------|
| OP-1      | Material | Strain/ Stress | 1,000 | 4,000 | OpenSEES wiki, 2021 |
| OP-2      |          |               |               |              |            |
| OP-3      |          |               |               |              |            |
| OP-4      |          |               |               |              |            |
| Huang     | Component | Displacement/ Reaction force | 2,000 | 320 | Uriz and Mahin, 2008 |
| Huang-N   |          |               |               | 40 40        | Huang, 2009 |
| BoucWen   | Structural | Ground Motion Acceleration/ Inter-story drift | 500 | 37 | Sun et al., 2013 |
|           |          |               |               | 13 50        | Zhang et al., 2019 |
| MRFDDBF   |          |               | 1,000 | 47 20 | Dong et al., 2018 |
|           |          |               |               | 481          | Zhang et al., 2019 |

3.1 Refined brace model

The refined finite element (FE) model of a brace component (Uriz and Mahin, 2008; Huang, 2009) is selected. The model is established based on LS-DYNA using refined-mesh shell elements and a cyclic damage plasticity model. This model can simulate brace behaviors under complex loads, including material nonlinearity (yield, strengthening, and softening), geometric nonlinearity (large deformation, buckling), damage, and fracture propagation. Consequently, this model has been widely adopted (e.g., Li et al., 2013; Amiri et al., 2013; Moharrami and Koutromanos, 2017; Lu and Guan, 2021), and it is suitable as a representative benchmark.

200 ground motion records with different characteristics are selected from the K-NET database (NIED, 2021). Their displacement sequences are obtained using the average acceleration integration method. The amplitudes of the displacement sequences range between
76.2 and 88.9 mm (i.e., 3.0 and 3.5 inch, approximately 25-30 times of the yield displacement of the brace) (Fig. 3(a)). In addition, based on several randomly selected sine waves, wave synthesis is conducted to construct 200 artificial displacement time series. Subsequently, the displacement sequences above are input into the LS-DYNA model, and 400 pairs of displacement-reaction force sequences are obtained. The amplitudes of the reaction force sequences are between 1081 and 1183 kN (243 and 267 klbf, Fig. 3(b)). The simulation continues from the elastic stage to the failure stage, which covers the entirety of the hysteretic behavior.

Based on the study of Zhang et al. (2019), the established dataset is normalized to the interval [-1, 1] through the min-max normalization. To further verify the robustness of the proposed method and analyze the effect of normalization, the original data without normalization are retained. All samples are randomly segregated into three datasets containing 320, 40, and 40 samples, which are used as training, validation, and testing datasets, respectively. The two datasets with normalized and original data are denoted as cases “Huang-N” and “Huang”, respectively.

3.2 Damped three-story frame

Dong et al. (2018) establish a three-story frame model. This frame can be categorized into a lateral resisting system, damping system, and gravity load system. The lateral resisting system includes eight identical single-bay moment-resisting frames (MRFs), and the damping system comprises eight single-bay frames with nonlinear viscous dampers and the associated bracing
(damped braced frame, DBF). The damping effect is significant in this case. Hereinafter, this case is referred to “MRFDBF”.

Based on the proposed model, Zhang et al. (2019) select 100 ground motions from the PEER NGA database (PEER, 2021) and perform the incremental dynamic analysis. Subsequently, a dataset with 548 samples is established and normalized. In this dataset, the ground motion acceleration and the inter-story drift (ISD) of each story are selected as the input and output, respectively. To prove the applicability of the redesigned network on small datasets, the dataset division method proposed by Zhang et al. (2019) is adopted. Only 47 samples are used for training, 20 samples are used for validation, and the remaining 481 samples are used for testing.

The amplitude distributions of the data are shown in Figs. 4(a-b). The output uniformly distributes in the range of 0.026 and 0.158 m, which corresponds to an inter-story drift ratio (IDR) of 1/124 to 1/20. Such a response distribution encompasses the elastic and nonlinear phases of the frame.

3.3 Giuffré-Menegotto-Pinto material cases with different characteristics

Based on the Giuffré-Menegotto-Pinto material model in OpenSEES, four different combinations of recommended parameters (OpenSEES wiki, 2021) are selected to construct datasets of material hysteretic behaviors with different characteristics. Specifically, ground motions are obtained based on the PEER NGA database (PEER, 2021) and the K-NET database (NIED, 2021) and pre-processed. Subsequently, the displacement sequences of those ground
motions are used as the inputs to calculate the hysteretic behaviors of different materials. The strain and stress of the materials are selected as the input and output of the network, respectively. Finally, the datasets are normalized using the aforementioned method and randomly segregated into three groups containing 4,000, 1,000, and 1,000 samples, which are regarded as the training, validation, and testing datasets, respectively.

The distribution of the peak strain is shown in Fig. 5. The responses encompass both the elastic and elastoplastic stages. Therefore, the obtained datasets are representative. These four cases are denoted as OP-1 to OP-4, respectively, which include different levels of hardening, softening, and deterioration.

3.4 BoucWen case

The BoucWen model (Bouc, 1967; Wen, 1980) is a widely used nonlinear hysteretic model. The basic principle of this model is shown in Fig. 6(a). Zhang et al. (2019) set different key parameters and select random band-limited white noise ground motion sequences with different magnitudes as inputs to construct a dataset containing 100 samples. A typical input sequence is shown in Fig. 6(b). The output is the ISD of a five degree-of-freedom system (Sun et al., 2013) based on the BoucWen model, and the normalization process is performed. Subsequently, the dataset is segregated into three groups containing 37, 13, and 50 samples, which are used as the training, validation, and testing datasets, respectively.
4. Discussion of Key Settings

4.1 Principle verification

As discussed in Section 2, the weighted pyramid network is designed based on the following two principles: (1) The positionings of features are varied, so different weights should be given in the feature fusion process; (2) Integrating multi-level features in the output module directly is beneficial for improving the final prediction performance.

To verify the rationality of these principles, a “separated” network is specially designed (as shown in Fig. 7). In Fig. 7, light blue arrows represent connections between layers that allow both forward and backpropagation, while black arrows represent connections that only allow forward operation and do not participate in the backpropagation of gradient information. In this network, it is necessary to cut off the gradient backpropagation between the feature extraction module and the prediction modules. If the backpropagation paths are not cut off, the "depth" of the features (that is, the number of layers between the selected hidden layer and the final output) calculated through different paths will be conflicted, making it difficult to effectively define and distinguish the "level" of each feature.

In this section, Cases Huang-N and MRFDBF are selected for verification. These two cases are derived from literatures and have been recognized in several follow-up studies. In addition, these relatively complex cases demonstrate consistency with practical applications. Therefore, the discussions based on these two cases are reliable. The model performances are
shown in Fig. 8 and Table 2. The loss used in this section is the mean-square error (MSE).

**Feature Extraction Module**

![Diagram of Feature Extraction Module]

**Fig. 7** The “separated” network for principle verification

**Fig. 8** Prediction performances based on different levels of features

(a) Extraction module: LSTM  
(b) Extraction module: Transformer

**Table 2** Comparison of the prediction performances

| Network for feature extraction | Case       | Minimum MSE based on single-level features | Minimum MSE based on multi-level features |
|-------------------------------|------------|------------------------------------------|------------------------------------------|
| LSTM                          | Huang-N    | 1.26×10⁻³                                | 1.24×10⁻³                                |
| LSTM                          | MRFDBF     | 4.44×10⁻³                                | 4.22×10⁻³                                |
| Transformer                   | Huang-N    | 2.61×10⁻⁴                                | 2.21×10⁻⁴                                |
| Transformer                   | MRFDBF     | 5.87×10⁻⁴                                | 3.33×10⁻⁴                                |
From Fig. 8 and Table 2, it can be seen that:

(1) As the depth of the hidden layer increases, the prediction performance gradually improves, which confirms the aforementioned principle 1 (features should be treated differently).

(2) The prediction results based on multi-level features are better than that based on single-level features, which confirms principle 2 (multi-level feature fusion in the output module directly is beneficial for prediction).

4.2 Comparison of key network settings

In the proposed weighted stacked pyramid architecture, various plans can be adopted to add shortcuts. Meanwhile, different values of the weight decay factor $p$ (in Equation (2)) can be selected. Therefore, analysis and comparison will be carried out in this section. Some of the common settings are as follows:

(1) The comparison and selection are conducted based on LSTM and Transformer. As mentioned above, cases Huang, Huang-N and MRFDBF are derived from existing literatures and have been recognized in several follow-up studies, thus will be used for comparison.

(2) Owing to the variance of performances on different datasets, the “normalized loss” is proposed to quantify the relative advantages of different schemes, as shown in Equation (3).

$$
Loss_N = \frac{Loss}{Loss_{max}}
$$

where $Loss_N$ represents the normalized loss. $Loss$ and $Loss_{max}$ represent the absolute and maximum values of validation loss obtained for comparison, respectively.

4.2.1 Comparison of different LSTM architectures

The following four LSTM architectures are established to discuss the effects of the number of LSTM layers and the manner by which the shortcuts are added:

(1) LSTM-1: one LSTM layer and no shortcut is available

(2) LSTM-2: two LSTM layers and a shortcut is added after the first layer

(3) LSTM-3: three LSTM layers and shortcuts are added after the first two layers

(4) LSTM-4: three LSTM layers and a shortcut is added after the second layer
As shown in Fig. 9(a), LSTM-2 performs the best; hence, it is selected for further comparison of different weight decay factors $p$. As shown in Fig. 9(b), when $p = 2.0$, the network exhibits the best performance.

4.2.2 Comparison of different Transformer architectures

Five different Transformer architectures are designed, and the features imported into the output module (through multi-level shortcuts) in each architecture are shown below:

1. Transformer-1 (TF-1): Decoder Outputs L3 and L4
2. Transformer-2 (TF-2): Decoder Outputs L2 and L4
3. Transformer-3 (TF-3): Decoder Outputs L2-L4
4. Transformer-4 (TF-4): Decoder Outputs L1-L4
5. Transformer-5 (TF-5): Decoder Outputs L1-L4; Encoder Outputs L1-L2

The comparison results are shown in Fig. 9(c), which confirms that importing more levels of features into prediction is conducive to improving the network performance. The Transformer-5 (TF-5) network performs the best under the current circumstances; hence, it is selected for further comparison of the weight decay factor $p$, as shown in Fig. 9(d). When $p = 2.0$, the network exhibits the best performance.

![Fig. 9 Comparison of different architectures and weight decay factors](image-url)
5. Network Testing and Comparison

The redesigned networks (Pyramid-LSTM, Pyramid-Transformer, and Pyramid-GA) are tested on all eight datasets and compared with several commonly used networks in this section.

5.1 Networks for comparison

As mentioned in the introduction, commonly used neural networks in hysteretic behavior simulation include MLP, LSTM/GRU, attention-based networks (like Transformer), the combination of LSTM/GRU with attention mechanism, and CNN. Therefore, the following networks proposed and adopted in existing studies are selected for comparison, which cover the aforementioned categories.

(1) Classical LSTM, Transformer, MLP, and 1D-CNN

These networks are widely used models and thus will no longer be explained here.

(2) Physics-guided CNN (named as “PhyCNN” hereafter)

PhyCNN is proposed by Zhang et al. (2020), which is regarded as one of the representative works of combining physical laws with neural networks. Additional constraints are added through the network architecture and loss function modification, which improves the network performance.

(3) Recursive LSTM (named as “Rec-LSTM” hereafter)

The Rec-LSTM network is an improved network proposed by Xu et al. (2022) based on classical LSTM. This network divides the time series into multiple windows at certain intervals, and predicts the response of the specified time step based on the information within a specific range (rather than using all historical time steps). This network is a representation of establishing local time-series correlation.

(4) Unrolled Attention Sequence-to-Sequence network (named as “UA” hereafter)

The UA network is a response time-series prediction network proposed by Wang et al. (2020). There are many similar studies combining the LSTM/GRU with the attention mechanism (e.g., Li et al., 2021a, 2021b), while their network architectures are basically the same, so the UA network is selected as the representation of these studies.
5.2 Network comparison results

In this section, the training and testing of the networks and the comparisons of different architectures are carried out. The hyperparameters of all networks are thoroughly adjusted, and the best performances are adopted for comparison. The performance of each network is shown in Table 3. Fig. 10 shows the average normalized loss on the testing datasets of all eight cases.

![Average normalized loss of each network (testing datasets)](image)

**Fig. 10** Average normalized loss of each network (testing datasets)

Since certain physical laws (such as the differential relationships) are necessary for the training process of the PhyCNN, the datasets constructed in Section 3 cannot fulfill the requirements. Therefore, the datasets and results provided by Zhang et al. (2020) when proposing the PhyCNN are adopted in comparison. Under the same physical constraints, the Pyramid-LSTM network is trained, and the results are shown in Table 4. It can be seen that the Pyramid-LSTM outperforms the PhyCNN on all three datasets. Due to the significant differences in data requirements, the PhyCNN network will not be further discussed.

From Tables 3-4 and Fig. 10, the following conclusions can be drawn:

(1) The weighted stacked pyramid network architecture achieves the best performance in all cases. Compared with other existing networks, the testing MSEs are reduced by 34.7% on average.

(2) Taking the testing normalized loss as the indicator, the Pyramid-GA network performs the best, followed by the UA network, the Pyramid-Transformer, and the Pyramid-LSTM.

(3) The basic network architecture has a great influence on the performance of the model. With the same basic architecture (for example, Pyramid-LSTM and LSTM have the same basic...
architecture), the redesigned weighted pyramid network outperforms the original network in 87.5% of tasks. The proposed multi-level shortcuts and weighted feature fusion mechanism are versatile in different basic architectures.

To show the prediction deviations more intuitively, Fig. 11 provides the prediction results and corresponding ground truth of the three proposed networks in the testing dataset. For brevity, only the results of some cases are given here. In Fig. 11, the orange lines represent the ground truth, and the blue lines represent the predicted results. It can be seen that the prediction results based on the weighted pyramid networks agree well with the ground truth.

![Fig. 11 Typical prediction results of the weighted pyramid networks](image-url)
Table 3 Network performances (testing MSE)

| Case     | LSTM   | Transformer | MLP   | UA   | 1D-CNN | Rec-LSTM | Pyramid-LSTM | Pyramid-Transformer | Pyramid-GA |
|----------|--------|-------------|-------|------|--------|----------|--------------|--------------------|------------|
| OP-1     | 2.73×10^{-7} | 4.60×10^{-6} | 5.81×10^{-5} | 8.90×10^{-8} | 6.88×10^{-7} | 5.98×10^{-6} | **3.86×10^{-8}** | 3.23×10^{-6} | 9.10×10^{-8} |
| OP-2     | 1.08×10^{-6} | 1.38×10^{-4} | 5.57×10^{-3} | 3.00×10^{-6} | 2.48×10^{-5} | 5.47×10^{-5} | **5.87×10^{-7}** | 1.17×10^{-4} | 2.10×10^{-6} |
| OP-3     | 1.28×10^{-6} | 1.53×10^{-4} | 6.26×10^{-3} | 3.53×10^{-6} | 3.34×10^{-5} | 1.56×10^{-4} | **9.40×10^{-7}** | 1.44×10^{-4} | 1.11×10^{-6} |
| OP-4     | 3.08×10^{-6} | 7.65×10^{-5} | 1.70×10^{-3} | 1.37×10^{-6} | 1.09×10^{-4} | 3.15×10^{-5} | 3.86×10^{-6} | 7.64×10^{-5} | **3.07×10^{-7}** |
| Huang-N  | 7.63×10^{-4} | 3.31×10^{-4} | 7.79×10^{-2} | 6.43×10^{-5} | 8.35×10^{-3} | 8.45×10^{-4} | 7.47×10^{-4} | 2.33×10^{-4} | **5.35×10^{-5}** |
| Huang    | 5.85×10^{-1} | 1.68×10^{-1} | 4.78×10^{1}  | 3.47×10^{0}  | 4.65×10^{2}  | 3.63×10^{1}  | 4.17×10^{1}  | 1.04×10^{1}  | **3.18×10^{0}** |
| MRFDBF   | 4.98×10^{-3} | 1.70×10^{-3} | 2.58×10^{-2} | 2.24×10^{-4} | 1.71×10^{-2} | 2.36×10^{-3} | 4.90×10^{-3} | 1.48×10^{-3} | **2.02×10^{-4}** |
| BoucWen  | 2.99×10^{-3} | 3.41×10^{-3} | 7.81×10^{-2} | 7.24×10^{-5} | 2.52×10^{-2} | 3.78×10^{-2} | 3.64×10^{-3} | 1.04×10^{-3} | **4.46×10^{-5}** |

Table 4 Comparison of the PhyCNN and Pyramid-LSTM

| Case     | num_ag2utt | exp_ag2utt |
|----------|------------|------------|
|          | PhyCNN     | Pyramid-LSTM | PhyCNN     | Pyramid-LSTM | PhyCNN     | Pyramid-LSTM |
| Loss 1   | 6.60×10^{-4} | **1.59×10^{-4}** | 9.43×10^{-4} | **2.19×10^{-4}** | 2.63×10^{-6} | **1.70×10^{-6}** |
| Loss 2   | 4.54×10^{-1} | **4.40×10^{-1}** | 2.41×10^{-2} | **7.11×10^{-3}** | 8.93×10^{-1} | **5.14×10^{-1}** |
| Loss 3   | -          | 7.07×10^{-1} | **2.82×10^{-1}** | -          | -          | -          |

Note: In case “num_ag2utt” and “exp_ag2utt”, Loss 1 and Loss 2 represent the deviations of the predicted acceleration and displacement sequences, respectively. In case “num_ag2u”, Loss 1, Loss 2, and Loss 3 represent the deviations of the predicted displacement, velocity, and acceleration sequences, respectively. Detailed information of these cases could be found in Zhang et al. (2020)
6. Conclusions

To further improve the reliability of hysteretic behavior simulations on various scales based on deep learning, a weighted stacked pyramid network architecture is proposed herein. The following conclusions are obtained:

(1) The proposed multi-level shortcut and weighted feature fusion strategy are reasonable. The testing MSEs of the proposed models are reduced by an average of 34.7% compared with existing widely-used networks. The prediction behaviors are consistent with the ground truth.

(2) Among all the network architectures discussed in this study, the proposed Pyramid-GA network performs the best, followed by the UA network, the Pyramid-Transformer, and the Pyramid-LSTM.

(3) The weighted pyramid architecture proposed in this study is versatile. If the basic architecture is the same, the proposed networks can achieve better performance in 87.5% of tasks.

Data Availability Statement

All datasets used in this study can be downloaded through: https://github.com/XYJ0904/Weighted-Pyramid-Stacked-Network. The codes and models will be uploaded after the publishing of this paper. Other data, models, and codes that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgment

The authors are grateful for the financial support from the National Key R&D Program (No. 2019YFC1509305) and Tencent Foundation through the XPLORER PRIZE, and Dr. Chen Wang in Tsinghua University for their kindly help.

References

Amiri, H. R., Aghakouchak, A. A., Shahbeyk, S., and Engelhardt, M. D. 2013. “Finite element simulation of ultra low cycle fatigue cracking in steel structures.” J. Constr. Steel Res.,
Bahdanau, D., Cho, K., and Bengio, Y. 2015. “Neural machine translation by jointly learning to align and translate.” In Proc., 3rd International Conf. on Learning Representations (ICLR), San Diego, California, USA.

Bouc, R. 1967. “Forced vibrations of mechanical systems with hysteresis.” In Proc., Fourth Conf. on Nonlinear Oscillations, Prague, Czech Republic.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 2014. “Learning phrase representations using RNN encoder-decoder for statistical machine translation.” In Proc., 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics.

Dong, B. P., Sause, R., and Ricles, J. M. 2018. “Seismic response and damage of reduced-strength steel MRF structures with nonlinear viscous dampers.” J. Struct. Eng., 144(12), 04018221. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002226.

Emelin, D., Titov, I., and Sennrich, R. 2019. “Widening the representation bottleneck in neural machine translation with lexical shortcuts.” In Proc., 4th Conference on Machine Translation (WMT), Florence, Italy: Association for Computational Linguistics.

Gorji, M. B., Mozaffar, M., Heidenreich, J. N., Cao, J., and Mohr, D. 2020. “On the potential of recurrent neural networks for modeling path dependent plasticity.” J. Mech. Phys. Solids, 103972. https://doi.org/10.1016/j.jmps.2020.103972.

Hao, J., Wang, X., Yang, B. S., Wang, L. Y., Zhang, J. F., and Tu, Z. P. 2019. “Modeling recurrence for transformer.” In Proc., 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, USA: North American Chapter of the Association for Computational Linguistics.

He, K. M., Zhang, X. Y., Ren, S. Q., and Sun, J. 2016. “Deep residual learning for image recognition.” In Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA: Institute of Electrical and Electronics Engineers.

Hochreiter, S., and Schmidhuber, J. 1997. “Long short-term memory.” Neural Comput., 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.
Huang, Y. L. 2009. “Simulating the inelastic seismic behavior of steel braced frames including the effects of low-cycle fatigue.” Doctoral Dissertation, University of California, Berkeley.

Kim, J., El-Khamy, M., and Lee, J. 2017. “Residual LSTM: Design of a deep recurrent architecture for distant speech recognition.” In Proc., 18th Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, Sweden: International Speech Communication Association.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. “ImageNet classification with deep convolutional neural networks.” Advances in Neural Information Processing Systems, 25, 1097-1105. https://doi.org/10.1145/3065386.

Li, G., Fahnestock, L. A., and Li, H. N. 2013. “Simulation of steel brace hysteretic response using the force analogy method.” J. Struct. Eng., 139(4), 526-536. https://doi.org/10.1061/(ASCE)ST.1943-541X.0000664.

Li, P. T. 2018. “Molecular dynamics study on the simulation model building and fracture mechanisms of metallic nano-materials.” Doctorial Dissertation, Northwestern Polytechnical University.

Li, T., Pan, Y., Tong, K., Ventura, C. E., and de Silva, C. W. 2021a. “A multi-scale attention neural network for sensor location selection and nonlinear structural seismic response prediction.” Comput. Struct., 248, 106507. https://doi.org/10.1016/j.compstruc.2021.106507.

Li, T., Pan, Y., Tong, K., Ventura, C. E., and de Silva, C. W. 2021b. “Attention-based sequence-to-sequence learning for online structural response forecasting under seismic excitation.” IEEE Trans. Syst. Man Cybern. Syst. 10.1109/TSMC.2020.3048696.

Liu, H., Liu, Z. Y., Jia, W. Q., Lin, X. K., and Zhang, S. 2020. “A novel transformer-based neural network model for tool wear estimation.” Meas. Sci. Technol., 31(6), 065106. https://doi.org/10.1088/1361-6501/ab7282.

Liu, Z. L., and Guo, A. X. 2021. “Empirical-based support vector machine method for seismic assessment and simulation of reinforced concrete columns using historical cyclic tests.” Eng. Struct., 237, 112141. https://doi.org/10.1016/j.engstruct.2021.112141.
Long, N. V. D., and Lee, M. 2012. “Dividing wall column structure design using response surface methodology.” *Comput. Chem. Eng.*, 37, 119-124. https://doi.org/10.1016/j.compchemeng.2011.07.006.

Lu, X. Z., Xu, Y. J., Tian, Y., Cetiner, B., and Taciroglu, E. 2021. “A deep learning approach to rapid regional post-event seismic damage assessment using time-frequency distributions of ground motions.” *Earthq. Eng. Struct. D.*, 50(6), 1612-1627. https://doi.org/10.1002/eqe.3415.

Lu, X. Z., and Guan, H. 2021. *Earthquake disaster simulation of civil infrastructures: From tall building to urban areas (2nd edition).* Singapore: Springer. https://doi.org/10.1007/978-981-15-9532-5.

Moharrami, M., and Koutromanos, I. 2017. “Finite element analysis of damage and failure of reinforced concrete members under earthquake loading.” *Earthq. Eng. Struct. D.*, 46(15), 2811-2829. https://doi.org/10.1002/eqe.2932.

NIED. 2021. “The NIED strong-motion seismograph networks.” Accessed on June, 2021. http://www.kyoshin.bosai.go.jp.

OpenSEES Wiki. 2021. “Steel02 Material - Giuffré-Menegotto-Pinto model with isotropic strain hardening.” Accessed on August, 2021. https://opensees.berkeley.edu/wiki/index.php/Steel02_Material--__Giuffr%C3%A9-Menegotto-Pinto_Model_with_Isotropic_Strain_Hardening.

Pacific Earthquake Engineering Research Center (PEER). 2021. “PEER NGA database (NGA-West, NGA-West2, NGA-East).” Accessed on June, 2021. https://peer.berkeley.edu/research/data-sciences.

Pham, A. D., Hoang, N. D., and Nguyen, Q. T. 2016. “Predicting compressive strength of high-performance concrete using metaheuristic-optimized least squares support vector regression.” *J. Comput. Civil Eng.*, 30(3), 06015002. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000506.

Redmon, J., and Farhadi, A. 2018. “Yolov3: An incremental improvement.” *arXiv preprint*. arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767.
Ronneberger, O., Fischer, P., and Brox, T. 2015. “U-net: Convolutional networks for biomedical image segmentation.” In Proc., 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany: Medical Image Computing and Computer Assisted Intervention Society.

Sun, H., Luş, H., and Betti, R. 2013. “Identification of structural models using a modified artificial bee colony algorithm.” Comput. Struct., 116, 59-74. https://doi.org/10.1016/j.compstruc.2012.10.017.

Tao, L., and Shahsavari, R. 2017. “Diffusive, displacive deformations and local phase transformation govern the mechanics of layered crystals: the case study of tobermorite.” Sci. Rep., 7(1), 5907. https://doi.org/10.1038/s41598-017-05115-4.

Tan, M. X., Pang, R. M., and Le, Q. V. 2020. “Efficientdet: Scalable and efficient object detection.” In Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), virtual-only conference: Institute of Electrical and Electronics Engineers.

Uriz, P. and Mahin, S. A. 2008. “Toward earthquake-resistant design of concentrically braced steel-frame structures.” Pacific Earthquake Engineering Research Center, Scientific Report, Berkeley, California, USA.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... and Polosukhin, I. 2017. “Attention is all you need.” In Proc., 31st International Conference on Neural Information Processing Systems (NeurIPS), Long Beach, USA: NeurIPS Foundation.

Wang, K., and Sun, W. C. 2018. “A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning.” Comput. Methods Appl. Mech. Eng., 334, 337-380. https://doi.org/10.1016/j.cma.2018.01.036.

Wang, C., Xu, L. Y., and Fan, J. S. 2020. “A general deep learning framework for history-dependent response prediction based on UA-Seq2Seq model.” Comput. Methods Appl. Mech. Eng., 372, 113357. https://doi.org/10.1016/j.cma.2020.113357

Wen, Y. K. 1980. “Equivalent linearization for hysteretic systems under random excitation.” J. Appl. Mech., 47(1), 150-154. https://doi.org/10.1115/1.3153594.

Xu, Y. J., Lu, X. Z., Tian, Y., and Huang, Y. L. 2020. “Real-time seismic damage prediction
Xu, Y. J., Lu, X. Z., Cetiner, B., and Taciroglu, E. 2021. “Real-time regional seismic damage assessment framework based on long short-term memory neural network.” *Comput.-Aided. Civ. Inf.*, 36(4), 504-521. https://doi.org/10.1111/mice.12628.

Xu, Z., Chen, J., Shen, J., and Xiang, M. 2022. “Recursive long short-term memory network for predicting nonlinear structural seismic response.” *Eng. Struct.*, 250, 113406. https://doi.org/10.1016/j.engstruct.2021.113406.

Zhang, R. Y., Chen, Z., Chen, S., Zheng, J. W., Büyüköztürk, O., and Sun, H. 2019. “Deep long short-term memory networks for nonlinear structural seismic response prediction.” *Comput. Struct.*, 220, 55-68. https://doi.org/10.1016/j.compstruc.2019.05.006.

Zhang, R. Y., Liu, Y., and Sun, H. 2020. “Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling.” *Eng. Struct.*, 215, 110704. https://doi.org/10.1016/j.engstruct.2020.110704.

Zhao, Y., Yang, R. N., Chevalier, G., Xu, X. M., and Zhang, Z. X. 2018. “Deep residual Bidir-LSTM for human activity recognition using wearable sensors.” *Math. Probl. Eng.*, 7316954. https://doi.org/10.1155/2018/7316954.