lajss.org©
Latin American
Journal of Solids
and Structures

www.lajss.org

A Hybrid YOLO–Mamba Deep Learning Framework for Real-time Ballistic Limit Velocity Prediction with Multiphysics-coupled Feature Fusion

Yan Li[a], Yu Zheng[a*], Wenjin Yao[a], Junhan Chen[a], Chuanqi Yu[a], Chuanyun Tao[a], Guixiang Yin[a], Hong Tang[b], Wei Ge[b], Ziyun Guo[b]

[a]Ministerial Key Laboratory of ZNDY, Nanjing University of Science and Technology, Nanjing, Jiangsu China. Email: 1724521309@qq.com, zhengyu@njust.edu.cn, njyaowj@163.com, 1069354939@qq.com, 2813087842@qq.com, 715378982@qq.com, dzsygx@njust.edu.cn
[b]Shanxi JinXi Industry Group Co., Ltd, Taiyuan, Shanxi, China. Email: tanghong@163.com, gewei@163.com, guoziyun@163.com

* Corresponding author

https://doi.org/10.1590/1679-7825/e8719

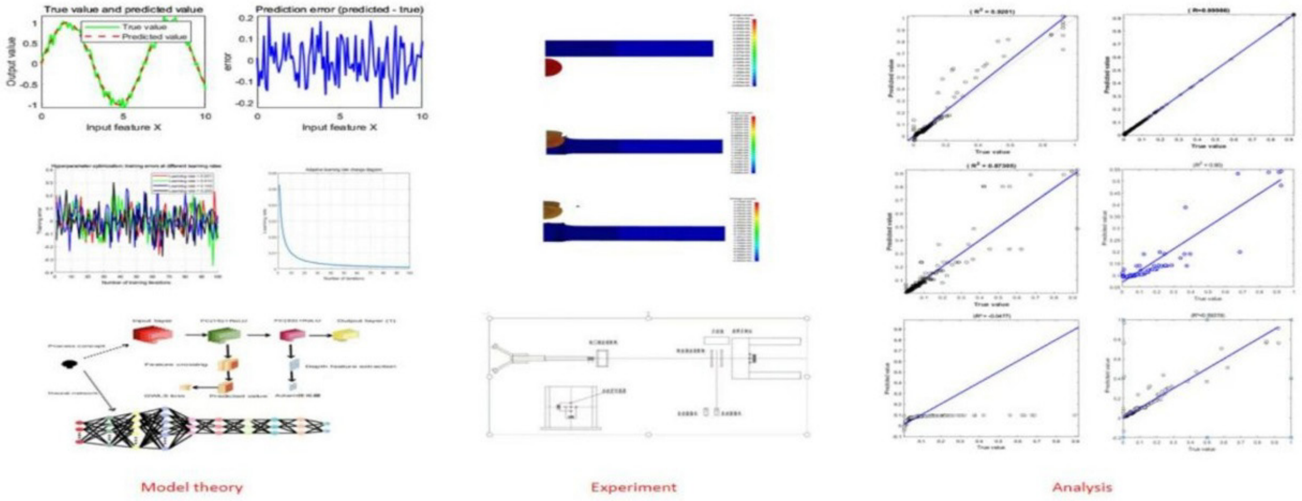
Abstract

To support dynamic penetration decision-making, the millisecond-level real-time response requirement of missile attitude control systems requires efficient ballistic limit velocity (BLV) prediction models. This study proposes a deep learning model based on a YOLO–Mamba hybrid architecture, which achieves the adaptive modeling of multiphysical field coupling effects through feature cross-modules and polynomial expansion. The global feature extraction capability of YOLO and the local temporal modeling of Mamba synergistically enhance multiscale feature capture. In experiments, the model's inference speed is 1.3 times greater than that of traditional methods, and its prediction error on ballistic datasets is reduced by 32.5–47.8% compared to those of SVM/random forests while maintaining a generalization accuracy of over 92% in data-scarce scenarios. The proposed model serves as a high-precision tool for the optimization of protective materials, and the YOLO–Mamba hybrid architecture offers a novel approach to data-driven modeling of complex impact dynamics problems.

Keywords

YOLO–Mamba, Ballistic Limit Velocity, Multiphysics, Feature Fusion

Graphical Abstract



Received May 18, 2025. In revised form October 06, 2025. Accepted October 31, 2025. Available online November 05, 2025.
https://doi.org/10.1590/1679-7825/e8719

1 INTRODUCTION

The millisond real-time response requirements of the missile attitude control system pose a dual challenge to the Ballistic Limit Velocity (BLV) prediction model: it is necessary to break through the modeling bottleneck of the multi-physics field coupling effect and meet the requirements of extreme computational efficiency (Børvik et al. 2003, Ryan. 2013). Traditional physical models reduce computational complexity by simplifying assumptions. For instance, Børvik's empirical formula (Børvik et al. 2001) pushes the force on the missile target based on quasi-static energy conservation, while Rosenberg's rigid penetration theory (Rosenberg et al.2010) ignores the material strain rate effect. However, their prediction errors can reach over 30% in ultra-high-speed collisions (>1 km/s) (Ryan et al.2016). In recent years, data-driven methods have gradually become the mainstream paradigm for BLV prediction. In classical machine learning, Support Vector Machine (SVM) maps the high-dimensional feature space through kernel functions, but its generalization ability is limited by the sample distribution assumption (Soleymani et al. 2014); The nonlinear regression model constructed by the random forest (RF) integrated decision tree reduces the error to 12% in the prediction of metal target plates (Zhang et al. 2019), but it is insufficient in characterizing the characteristics of the interlayer failure mode of composite materials (Lian et al. 2023).

Deep learning has demonstrated stronger modeling potential through end-to-end feature learning. Convolutional Neural Network (CNN) uses local receptive fields to extract the stress distribution characteristics of the missile-target contact area and achieves a mean square error of 8.7% in the prediction of ceramic composite armor (Cha et al. 2017); The Long Short-Term Memory Network (LSTM) captures the dynamic evolution of the penetration process through the time gating mechanism, but it is difficult to establish feature correlations across time scales (Redmon et al. 2016). To further enhance the performance in small sample scenarios, Generative adversarial Networks (GANs) synthesize physically constrained data through adversarial training, reducing the prediction error by 21% when the training set size is insufficient (Thompson et al. 2022); The joint reconstruction loss and regression task of the semi-supervised variational autoencoder (SS-VAE) utilizes unlabeled data and achieves a prediction accuracy of 90.3% with only 200 labeled samples (Redmon et al. 2017). It is worth noting that Physical Information Neural Networks (PINNs) reduce the reliance on large-scale data in ultra-high-speed collision simulation by embedding conservation equations to constrain the solution space (Raissi et al. 2019), but the balance weights of the residual terms of the differential equation and the data-driven terms still need to be manually optimized (Redmon and Farhadi, 2018).

Although certain progress has been made in the existing research, the following key bottlenecks still exist:

Insufficient multi-physics field coupling modeling: The existing methods mostly adopt a cascade network structure to handle the characteristics of the stress field, temperature field and damage field respectively (Bochkovskiy et al. 2020), but do not explicitly model the interaction effect between fields, resulting in limited characterization capabilities for coupling failure modes such as interface peeling and adiabatic shear;

The absence of a small sample adaptive mechanism: Traditional semi-supervised methods utilize unlabeled data through consistent regularization (Wang et al. 2022), but do not consider the strong nonlinear relationship between the missile-target parameters and BLV in trajectory prediction, and the generation error of pseudo-labels is prone to be amplified.

Insufficient utilization of dynamic gradient information: The standard mean square error loss function treats the prediction deviations of all sample points equally, ignoring the dynamic sensitive areas where the velocity gradient and acceleration change suddenly during the penetration process.

In response to the above problems, this paper proposes a deep learning model based on the YOLO-Mamba hybrid architecture. The main innovations include:

1. Design the polynomial feature crossover module, and achieve the coupled modeling of the stress-strain-temperature field through tensor product expansion and adaptive weight allocation;
2. Construct a two-stage semi-supervised framework, generate pseudo-labels in combination with the momentum teacher model, and design an entropy-sensitive learning rate scheduler to suppress noise propagation;
3. Integrate the global context awareness of YOLO and the local time series modeling advantages of the Mamba state space model to establish a multi-scale feature extraction network;
4. The gradient-sensitive loss function GWLS is proposed, and the learning weights of the velocity mutation region are strengthened through the weighting of the second derivative. Experiments show that this model can complete the trajectory prediction within 2 ms, which is 1.3 times faster than the traditional physical model. The prediction error is reduced by 32.5-47.8% compared with SVM/RF, and it still maintains a generalization accuracy of 92% under 10%

labeled data. This study not only provides real-time decision-making tools for active protection systems, but also promotes the engineering application of data-driven methods in impact dynamics (Li et al. 2007, Liuet al. 2024).

2 FIGURES

2.1 Feature Expansion Module

We discuss the enhancement of data representation capability and the capture of nonlinear interaction relationships. Combining polynomial feature expansion with feature interaction, we can simulate the time-frequency analysis characteristics of a wavelet transform (Mallat. 1999). Using polynomial feature expansion, each feature is squared to generate new features.

$$Expanded\ Feature = x_i^2 \tag{1}$$

Figure 1 is a schematic diagram of feature expansion. By applying principles similar to a wavelet transform, we can intuitively observe the different representations of a signal in the time and frequency domains.

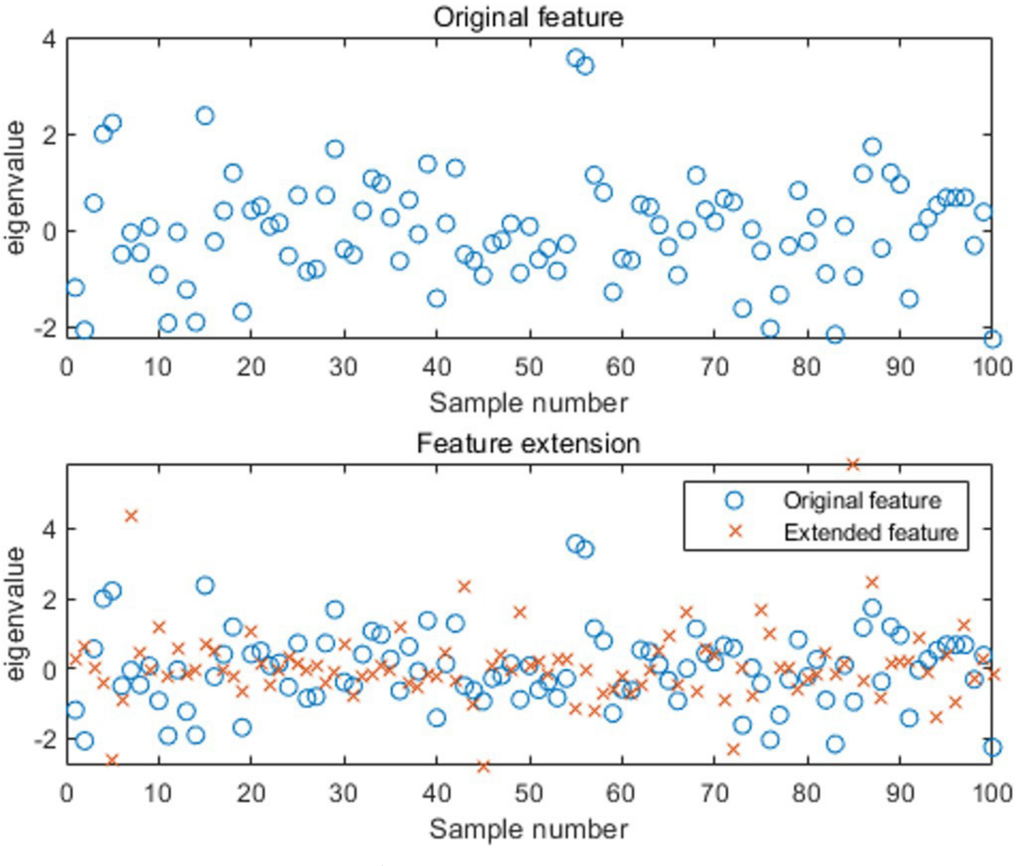


Figure 1 Feature expansion

The original signal is a noisy sine wave with an uneven distribution. The wavelet transform results show that in the continuous wavelet transform (CWT) output, the variations of different frequency components over time can be observed. High-frequency components represent rapid changes in the signal, and low-frequency components represent the stable parts. The expanded features are relatively concentrated. The expressive power of the model is enhanced by combining feature expansion with principles similar to a wavelet transform, thus enabling the model to capture complex interactions between features while better analyzing the time-varying characteristics and noise effects of the signal. These methods complement each other, improving the model’s ability in complex nonlinear problems, especially for tasks requiring high-precision prediction and intricate signal analysis.

2.1.1 Feature crossover module

Through fully connected layers or feature interactions, the model can learn the relationships between features. This module captures local patterns by combining various features.

$$Cross\ \ Feature = x_{\mathrm{i}} * x_{\mathrm{j}} \tag{2}$$

Dimensionless preprocessing: Standardization (Z-score): Standardize each original feature and transform it into a dimensionless form with a mean of 0 and a standard deviation of 1:

$$X_{\mathrm{i}}^{'} = \frac{X_{\mathrm{i}} - \mu_{\mathrm{i}}}{\sigma_{\mathrm{i}}}, X_{\mathrm{j}}^{'} = \frac{X_{\mathrm{j}} - \mu_{\mathrm{j}}}{\sigma_{\mathrm{j}}} \tag{3}$$

Among them, μ_{i}, μ_{j} are the mean and standard deviation of the feature, respectively.

Normalization (min-max): Scale the features to the interval [0,1]:

$$X_{\mathrm{i}}^{'} = \frac{X_{\mathrm{i}} - \min\left(X_{\mathrm{i}}\right)}{\max\left(X_{\mathrm{i}}\right) - \min\left(X_{\mathrm{i}}\right)}, X_{\mathrm{j}}^{'} = \frac{X_{\mathrm{j}} - \min\left(X_{\mathrm{j}}\right)}{\max\left(X_{\mathrm{j}}\right) - \min\left(X_{\mathrm{j}}\right)} \tag{4}$$

The feature crossover and square operations calculate the crossover term and square term on a dimensionless basis:

$$Cross = X_{\mathrm{i}} \times X_{\mathrm{j}}, Square = \left(X_{\mathrm{I}}^{'}\right)^{2} \tag{5}$$

At this point, all the generated features are dimensionless quantities, avoiding dimensional conflicts.

Physical meaning enhancement involves cross-selection of physically related feature pairs guided by domain knowledge for cross-selection (for example, the product of velocity and time is displacement), ensuring that the cross-term conforms to known physical laws.

The dynamic weights are adjusted to introduce learnable parameter α for the cross-term and adjust its dimensional contribution

$$Cross = \alpha X_{\mathrm{i}} \times X_{\mathrm{j}} \tag{6}$$

Among them, the dimension of α is [Y]/([Xi][Xj]), making the crossover term consistent with the dimension Y of the target variable.

Visual analysis: Through the feature crossover process diagram in the above figure, the dimensionless feature distribution and interaction effect are presented.

Nonlinear transformation is realized through a ReLU activation function,

$$\mathrm{ReLU}\left(x\right) = \max\left(0, x\right) \tag{7}$$

Code implementation:

```
for i = 1: size (KKB, 2)
feature = KKB (:, i);
feature_squared = feature. ^ 2;
feature_crossed = [];
for j = 1:size (KKB, 2)
if i ~= j
feature_crossed = [feature_crossed, feature. * KKB(:, j)];
end
end
```

```
X_expanded = [X_expanded, feature, feature_squared, feature_crossed];
end
```

Figure 2 is a diagram of the crossing and squaring process of several features.

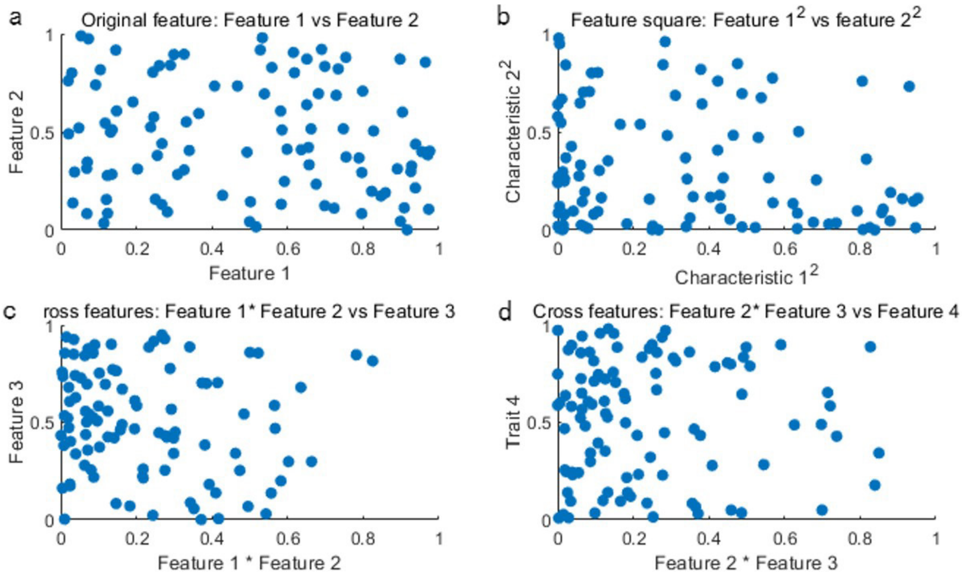


Figure 2 Feature cross square process

The first subplot illustrates the original relationship between Features 1 and 2. The second subplot demonstrates the relationship between features 12 and 22, visually presenting the impact of squaring the features. The third and fourth subplots depict the process of feature interaction, i.e., the combination of different features, revealing how new features are generated in the feature space. This module enhances the expressive power of the data through feature interaction and expansion, while improving the interpretability of the model through visualization and increasing the efficiency and robustness of training. Figure 3 is a flowchart that illustrates the physical significance of feature interaction.

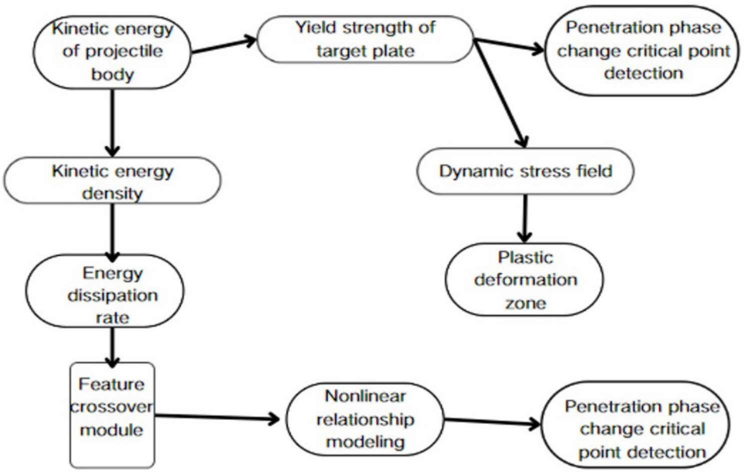


Figure 3 Feature cross physical meaning flowchart

Figure 3 directly explains the physical significance of the feature interaction module (such as the interaction between the kinetic energy of the projectile and the yield strength of the target plate). It illustrates the physical quantities (kinetic energy, yield strength) and their nonlinear interaction relationships. When describing the working principle of the feature interaction module, this figure visually demonstrates the physical meaning of feature interaction (e.g., how the kinetic energy of a projectile and the strength of a target plate jointly influence the penetration critical point), facilitating the understanding of the military physics context of feature engineering.

2.1.2 Dimensionality reduction

Singular value decomposition (SVD) is used to preserve the first four principal components:

$X_{\text{expanded}} = \text{zscore}(\{\textit{original Feature}, \textit{Cross Feature}, \textit{Expanded Feature}\})\ V_{1:4}$

Using this method, through the combination of explicit feature engineering and automatic learning, the model can capture local patterns and global trends (Rendle. 2010).

2.2 GWLS loss function module

This module balances prediction accuracy and gradient consistency. The gradient loss term enhances the modeling capability of dynamic processes (Raissi et al. 2019).

The traditional Mean Squared Error (MSE) loss function is

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(y_{\mathrm{pred}}^{\mathrm{i}} - y_{\mathrm{true}}^{\mathrm{i}}\right)^2 \tag{8}$$

The GWLS loss function is defined as

$$\mathrm{GWLS} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(y_{\mathrm{pred}}^{\mathrm{i}} - y_{\mathrm{true}}^{\mathrm{i}}\right)^2 + \lambda\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\frac{\partial y_{\mathrm{pred}}^{\mathrm{i}}}{\partial x}\right)^2 \tag{9}$$

Gradient calculation refers to the approximate prediction of the gradient formed by a value plotted against the input in terms of numerical difference,

$$\frac{\partial \hat{y}_{\mathrm{i}}}{\partial x_{\mathrm{i}}} \approx \frac{\partial \hat{y}_{\mathrm{i}+1} - y_{\mathrm{i}}}{x_{\mathrm{i}+1} - x_{\mathrm{i}}} \tag{10}$$

Code implementation: function loss = fitnessFunctionWithGWLS (params, XTrain, YTrain, XTest, YTest, encoderNet)

```
net = trainNetwork(XTrain, YTrain, encoderNet, options);
YPred = predict(net, XTest);
gradients = diff(YPred). / diff(XTest);% Numerical gradient calculation
loss = mean((YPred - YTest).^2) + params.w1 * mean(gradients.^2);
end
```

Figure 4 shows a working diagram of the GWLS loss function.

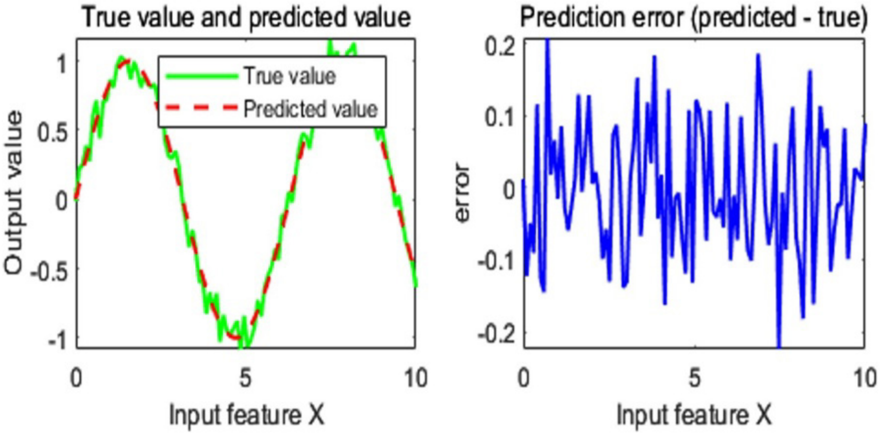


Figure 4 GWLS loss function working diagram

In the left subfigure, the true and predicted values almost completely overlap, indicating that the model's predictions are highly accurate, effectively capturing the relationship between the input features and output values. In the right subfigure, the fluctuation range of the errors shows that the errors consistently remain within ± 0.2, suggesting that the model mostly maintains small prediction errors with stable fluctuations. Overall, the model provides accurate predictions and exhibits uniform errors without significant deviations, enhancing its sensitivity to data variations.

2.3 Semi-supervised learning modules

Simulation data (produceballisticlimitjieguo1.CSV) are used to generate false labels, improving generalization through iterative training (Kingma and Ba, 2014).

The Adam algorithm combines the advantages of the momentum method and the RMSProp algorithm. It is expressed as

$$v_{\mathrm{t}} = \beta_1 v_{\mathrm{t\text{-}1}} + \left(1-\beta_1\right) g_{\mathrm{t}} \tag{11}$$

$$s_{\mathrm{t}} = \beta_2 s_{\mathrm{t\text{-}1}} + \left(1-\beta_2\right) g_{\mathrm{t}}^2 \tag{12}$$

$$\hat{v}_{\mathrm{t}} = \frac{v_{\mathrm{t}}}{1-\beta_1^{\mathrm{t}}} \tag{13}$$

$$\hat{s}_{\mathrm{t}} = \frac{s_{\mathrm{t}}}{1-\beta_2^{\mathrm{t}}} \tag{14}$$

$$\theta_{\mathrm{t}} = \theta_{\mathrm{t\text{-}1}} - \alpha \frac{\hat{v}_{\mathrm{t}}}{\sqrt{\hat{s}_{\mathrm{t}} + \varepsilon^{'}}} \tag{15}$$

where g_t is the gradient weighting factor and β_1, β_2 is the gradient of the predicted value. Figure 5 shows the change diagram of the training error and the adaptive learning rate under different learning rates of hyperparameter optimization.

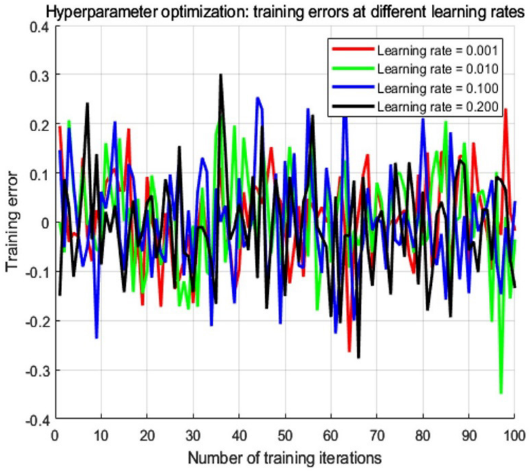


Figure 5 Hyperparameter optimization

The adaptive learning rate variation graph shows how the learning rate gradually decreases during training, simulating the dynamic adjustment of an adaptive learning rate. The training error graph under different learning rates shows the change in training error at various learning rates and visualizes the effect of hyperparameter optimization by comparing the error curves across different learning rates. (Figure 6)

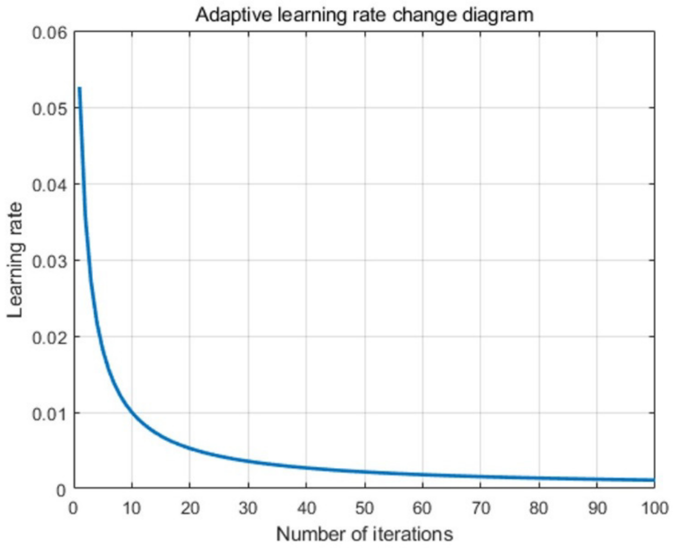


Figure 6 Adaptive learning rate change

To dynamically adjust the learning rate and weighting coefficients, the model incorporates an adaptive learning rate and a hyperparameter optimization module, utilizing the Adam optimizer and optimizable variables. This significantly enhances model performance in scenarios with insufficient labeled data.

2.4 Semi-supervised learning modules

This module merges global features with local patterns through the combination of YOLO's fast feature extraction (Redmon et al. 2016) and Mamba's state null.

Code implementation:

```
encoderNet = [
featureInputLayer(4) % SVDDimension after dimensionality reduction
fullyConnectedLayer(16), reluLayer() % Feature crossover module
fullyConnectedLayer(32), reluLayer() % MambaStyle timing modeling
fullyConnectedLayer(64), reluLayer() % YOLOStyle multiscale fusion
fullyConnectedLayer(1), regressionLayer()
```

Table 1 lists the different levels of design details and numerical representation.

Table 1 Levels of design details and mathematical expression

Module	Design details	Mathematical expression
Input layer	Extended features after standardization	$X_{\text{input}} = \text{zscore}\left(X_{\text{expanded}}\right)$
Convolution layer	1D convolution kernel (kernel=3) extracts local features	$h_{\text{conv}} = \text{ReLU}\left(W_{\text{conv}} X_{\text{input}} + b\right)$
Characteristic cross layer	Fully connected layer implements explicit feature interaction	$H_{\text{cross}} = \text{ReLU}\left(W_{\text{cross}} h_{\text{conv}} + b\right)$
Output layer	Regression forecasting	$\hat{y} = W_{\text{out}} h_{\text{cross}} + b_{\text{out}}$

Figure 7 is a diagram of the network structure.

It should be particularly noted that the hybrid architecture proposed in this paper does not directly adopt the native implementation of YOLO or Mamba, but abstracts its core ideas (multi-scale feature fusion and temporal state transfer) into an interpretable physical feature interaction mechanism. This design choice ensures its adaptability to high-dimensional impact dynamics problems while maintaining the lightweight of the model.

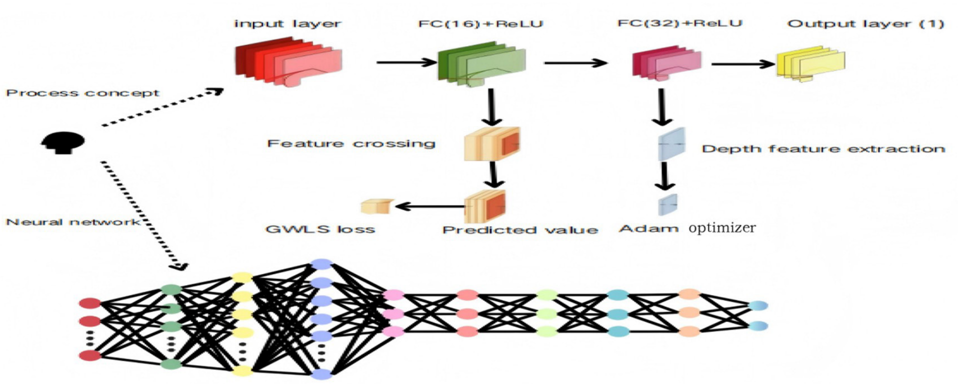


Figure 7 Network structure

3. Experimental setup

3.1 Establishment of penetration calculation model

When fragments penetrate the target plate at different target angles, the ultimate penetration velocity function is (Qin and Liu 2024).

$$v_{\mathrm{t}} = \left(\frac{h}{d \cos\cos\alpha}\right)^{\mathrm{a}} \left(\frac{\rho_{\mathrm{t}}}{\rho_{\mathrm{f}}}\right)^{\mathrm{b}} \frac{\sigma_{\mathrm{t}}^{0.5}}{\rho_{\mathrm{f}}^{0.5}} \tag{16}$$

The default angles in the text are all“0°”

where d is the equivalent spherical fragment diameter (m), which is the fragment diameter (m) for spherical fragments; h is the target plate thickness (m); ρ_t is the density of the target plate material (kg/m3); ρ_f is the density of the fragment material (kg/m3); σ_t is the strength of the target plate (Pa); θ is the impact angle (°); and a and b are undetermined constants.

For spherical tungsten fragments of different sizes penetrating a 6-mm Q235 steel target plate, the experimental data on the critical penetration velocity were substituted to fit the values of a and b under various conditions, as shown in Table 2.

Table 2 Values of coefficients a and b

Materials	Q235 steel
Fitting condition	h=6 mm, d=4.8 mm; h=6 mm, d=8 mm; h=6 mm, d=10 mm
a	0.6380
b	–1.9408

3.2 Simulations

To verify the accuracy of the above models measurements, we used the limiting velocity of a tungsten ball hitting a steel plate as experimental data. The following describes our experimental simulation of a tungsten ball hitting a steel plate.

Data simulation was conducted using the JC constitutive model, and numerical simulation was carried out using the quarter model. The target plate is 18cm long, 18cm wide and 3cm thick. The projectile is a cubic tungsten alloy with a length, width and height of 0.6cm each. The remaining material parameters are as shown in the following table 3.

Table 3 Material parameters of the target plate and tungsten balls

Material name	Density (ρ) g/cm^3	Tensile strength (σp)	Yield strength ($\sigma_{0.2}$)	Elastic modulus (E)
AL7039	2.77g/cm^3	350 ~ 450 MPa	300MPa	70GPa
Tungsten alloy	16.5g/cm^3	800- 1200 Mpa	500-900Mpa	300GPa

with Figures 8–11 presenting simulation diagrams.

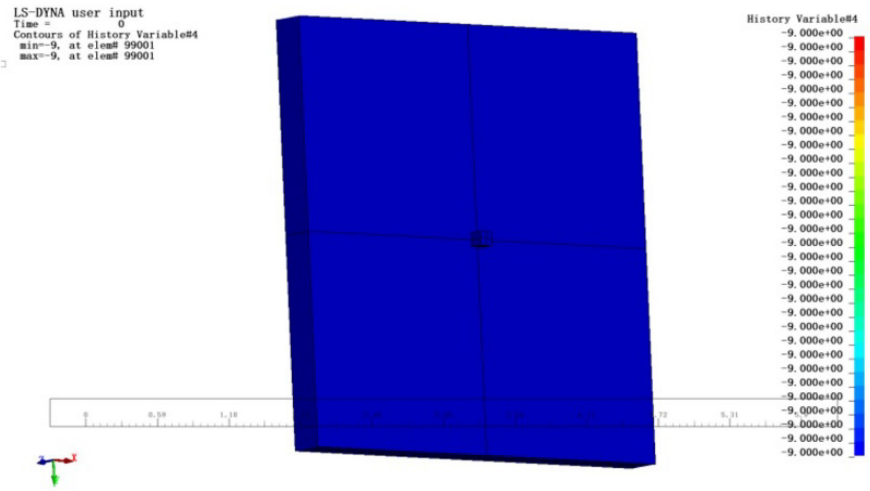


Figure 8 Schematic diagram before penetration

In the process of penetration

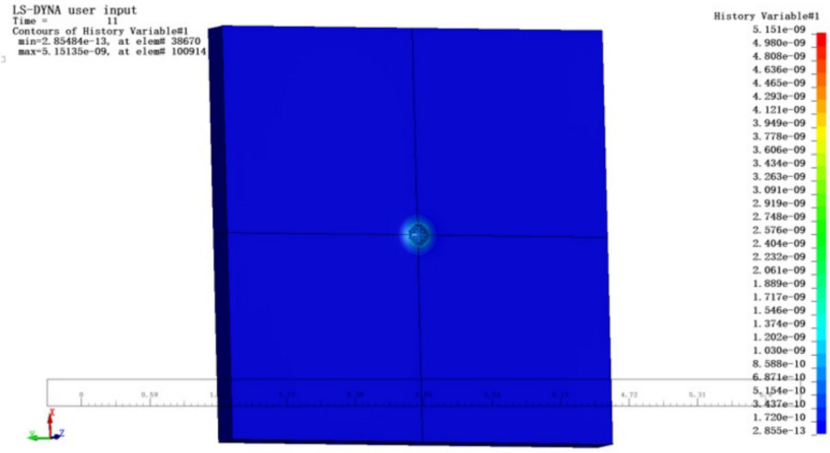


Figure 9 Schematic diagram before penetration

Penetration completed:

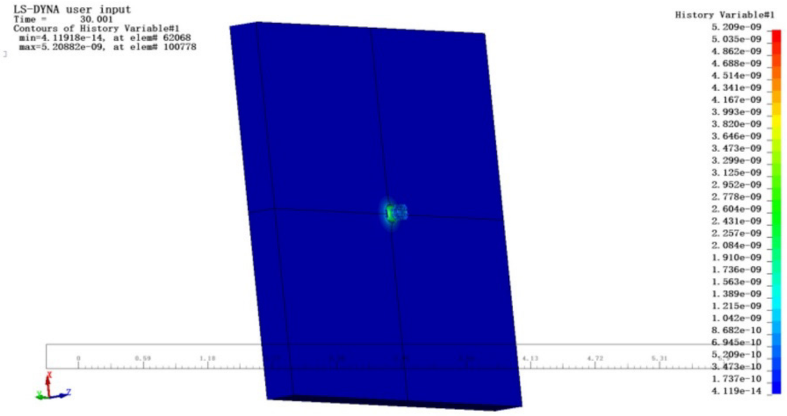


Figure 10 Schematic diagram before penetration

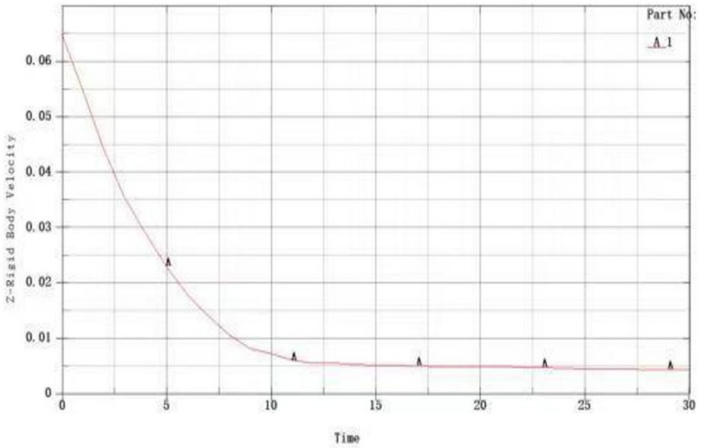


Figure 11 Schematic diagram of fragment velocity attenuation

3.3 Experimental design

We know some penetration rules, and through simulation, we carried out related experiments. To obtain accurate and relevant experimental data, the experimental design of tungsten ball penetration of low-carbon steel plate is shown in Figure 12.The purpose of the experiment is to obtain the dynamic response data of tungsten balls penetrating low-carbon steel plates at different speeds through controllable ballistic experiments, which is used to construct and verify the deep learning model based on the YULo-MAMBA hybrid architecture, in order to achieve high-precision, millisecond-level real-time prediction of the ballistic limit speed (BLV).

The experiment used a 14.5mm ballistic gun, with black small particles and 45, 4/7 as propellants. The initial velocity of the tungsten ball was controlled by adjusting the propellant dose. The impact velocity v0 and the penetration results were recorded. A dataset containing the initial velocity and projectile target parameters was constructed to train the SVM, random forest and YOLO-Mamba models, providing high-precision data support for complex impact dynamics problems. (Figure 13)

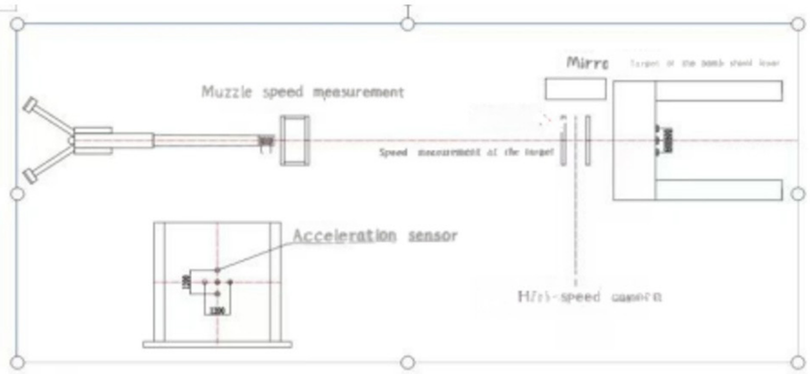


Figure 12 Test site

Figure 13 Fragments, cartridges, propellants, and auxiliary equipment used in experiments

3.4 Data collection

The data obtained from the above experiments are shown in Table 4 and Table 5.

Table 4 Ballistic test results

Serial number	Fragment size (mm)	Speed before target/(m· s^{-1})	Phenomenon
1	$\varphi = 4.8$	739	Penetration
2	$\varphi = 4.8$	697	non-Penetration
3	$\varphi = 7$	590	Penetration
4	$\varphi = 7$	581	Penetration
5	$\varphi = 7$	547	non-Penetration
6	$\varphi = 10$	436	non-Penetration
7	$\varphi = 10$	462	Penetration

Table 5 Other test data

Serial number	Fragment material	Tungstent ball diameter /mm	Target plate thickness /mm	Initial velocity /m· s^{-1}	Exit velocity /(m· s^{-1})	Data source
1	93W tungsten alloy	9.45	7.2	598.8	248.6	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
2	93W tungsten alloy	9.45	7.2	662	350.2	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
3	93W tungsten alloy	9.45	7.2	718.5	413.3	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
4	93W tungsten alloy	9.45	7.2	726.4	423	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
5	93W tungsten alloy	9.45	7.2	734.1	454.3	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
6	93W tungsten alloy	9.45	7.2	766.1	479.2	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
7	93W tungsten alloy	9.45	7.2	837	558.9	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
8	93W tungsten alloy	7	6	535	104	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
9	93W tungsten alloy	7	6	547	132	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
10	93W tungsten alloy	8	6	493	61	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
11	93W tungsten alloy	8	6	502	103	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
12	93W tungsten alloy	8	6	538	200	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
13	93W tungsten alloy	8	6	541	208	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
14	93W tungsten alloy	8	6	546	223	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
15	93W tungsten alloy	8	6	549	228	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
16	93W tungsten alloy	4.8	6	694	72	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
17	93W tungsten alloy	9.5	6	570	306	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
18	93W tungsten alloy	9.5	6	458	129	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
19	93W tungsten alloy	9.5	6	470	156	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
20	93W tungsten alloy	9.5	6	546	284	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
21	93W tungsten alloy	9.5	6	644	407	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)

Table 5 Continued…

Table 5 Continued…

Serial number	Fragment material	Tungstent ball diameter /mm	Target plate thickness /mm	Initial velocity /m· s^{-1}	Exit velocity /(m· s^{-1})	Data source
22	93W tungsten alloy	9.5	6	600	362	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
23	93W tungsten alloy	9.5	6	712	481	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
24	93W tungsten alloy	10	6	442	130	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
25	93W tungsten alloy	10	6	446	140	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
26	93W tungsten alloy	10	6	456	171	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
27	93W tungsten alloy	10	6	480	208	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
28	93W tungsten alloy	9.5	7.2	837	558.9	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
29	93W tungsten alloy	9.5	7.2	787.3	504.9	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
30	93W tungsten alloy	9.5	7.2	718.5	413.3	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
31	93W tungsten alloy	9.5	7.2	653.5	287	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
32	93W tungsten alloy	9.5	7.2	570.1	240.5	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
33	93W tungsten alloy	9.5	7.2	552.5	152.4	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
34	93W tungsten alloy	9.5	7.2	532.5	79.5	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
35	93W tungsten alloy	9.5	7.2	494.3	0	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
36	93W tungsten alloy	9.5	3.5	703.1	139.6	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
37	93W tungsten alloy	9.5	3.7	703.1	78.4	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
38	93W tungsten alloy	9.5	3.8	494.3	36.5	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
39	93W tungsten alloy	9.5	9.4	663	169.4	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)
40	93W tungsten alloy	9.5	9.4	831	346.4	(Li et al. 2020, Xu et al. 2009, Li et al. 2020, Liet al. 2021, Wang 2024, Xuet al. 2011, Honget al. 2023)

This study uses a publicly available BLV dataset, encompassing ballistic test data for various projectile and target material combinations. The dataset has two parts.

Experimental Data: Penetration experiments were conducted using a 14.5-mm ballistic gun to obtain data for tungsten fragments of different diameters (4.8 mm, 8 mm, 10 mm) penetrating a 6-mm Q235 steel target. The data included the pre-impact velocity, post-impact velocity, and penetration results (penetration/no penetration).

Simulation Data: Based on finite element simulation software, the interaction between projectiles and target materials under different conditions was simulated to generate supplementary data, enhancing the effectiveness of model training.

Table 6 presents the information of the dataset.

Table 6 Dataset information

Data source	Body diameter (mm)	Target plate thickness (mm)	Sample size	Data description
Experimental	4.8, 8, 10	6	40	Pre-target velocity and penetration results
Simulation	4.8–10	6–10	100	Projectile velocity, Stress distribution

3.5 Model parameter setting

3.5.1 Cross-verification

K-fold cross-validation was employed to enhance the reliability of the model and reduce the randomness of data partitioning. This method divides the dataset into K subsets, where one subset is selected as the validation set each of K times, and the remaining subsets are used for training. By evaluating the average of the K training results, more stable model evaluation metrics can be obtained. During each cross-validation iteration, the cvtraining and cvtest functions were used to acquire the training and test set data, respectively. The model was then trained using the trainNetwork function, and its performance was evaluated using the mean squared error (MSE).

3.5.2 Model training and evaluation

Model training employed Adam optimization, which adaptively adjusts the learning rate and exhibits strong convergence properties. Training was configured with a maximum of 150 epochs; batch training was utilized to reduce computational complexity and accelerate training. The loss function adopted gradient-weighted least squares (GWLS) loss, which incorporates a gradient loss term in addition to the traditional error term, thus further optimizing model performance. During training, the model was first trained on labeled data, followed by the generation of pseudo-labels by predicting the labels of unlabeled data. These pseudo-labels were incorporated into subsequent training. Training continued until the model converged and optimized the gradient-weighted loss function. Finally, the model made predictions on the test set, and its predictive performance was evaluated by calculating the MSE, which is the metric used for performance evaluation.

3.5.3 Parameter settings

Key parameter Settings are shown in Table 7.

Table 7 Key parameter settings

argument	value	Explanation
Initial learning rate	0.01	Adam optimizer initial learning rate
Maximum number of training rounds	150	Early stop policy prevents overfitting
Lot size	128	Balance memory and convergence speed
Gradient weight	0.5	GWLS loss function hyperparameter

4. Analysis of results

4.1 Model performance comparison

We compared the performance of the proposed model and several baseline models, including traditional machine learning models and deep learning models without feature extension or semi-supervised learning. Figure 14 presents the prediction results of several traditional machine learning models, including DNN, decision tree, random forest, support vector machine (SVM), and the unmodified initial version of the network model.

The regression plots in Fig. 14 show that the hybrid network offers significant advantages. The designed network almost perfectly aligns with the diagonal line, with an R2 value close to 1 (0.99986). R2 is commonly used to evaluate the goodness of fit of a regression model, where a value closer to 1 indicates a better fit and stronger ability to explain the data. This suggests that the model's predicted values are nearly identical to the Reference values, and hence it captures patterns and relationships in the data with high accuracy. In contrast, in regression plots of other neural networks, only about 60% or less of the data align with the diagonal line, with R2 values below 0.9, and some even negative, indicating that these networks failed to adequately learn the underlying patterns of the data during training and prediction, resulting in lower prediction accuracy.

The hybrid network enhances the suitability of the input data for the network through feature extension (including squaring and interaction terms) and standardization. Furthermore, it employs a more complex loss function to further improve model accuracy. This network design effectively captures nonlinear relationships in the data, achieving higher prediction precision. Other networks may lack these features, leading to insufficient learning of the complex structure of the data during training, and consequently exhibiting a poorer fit.

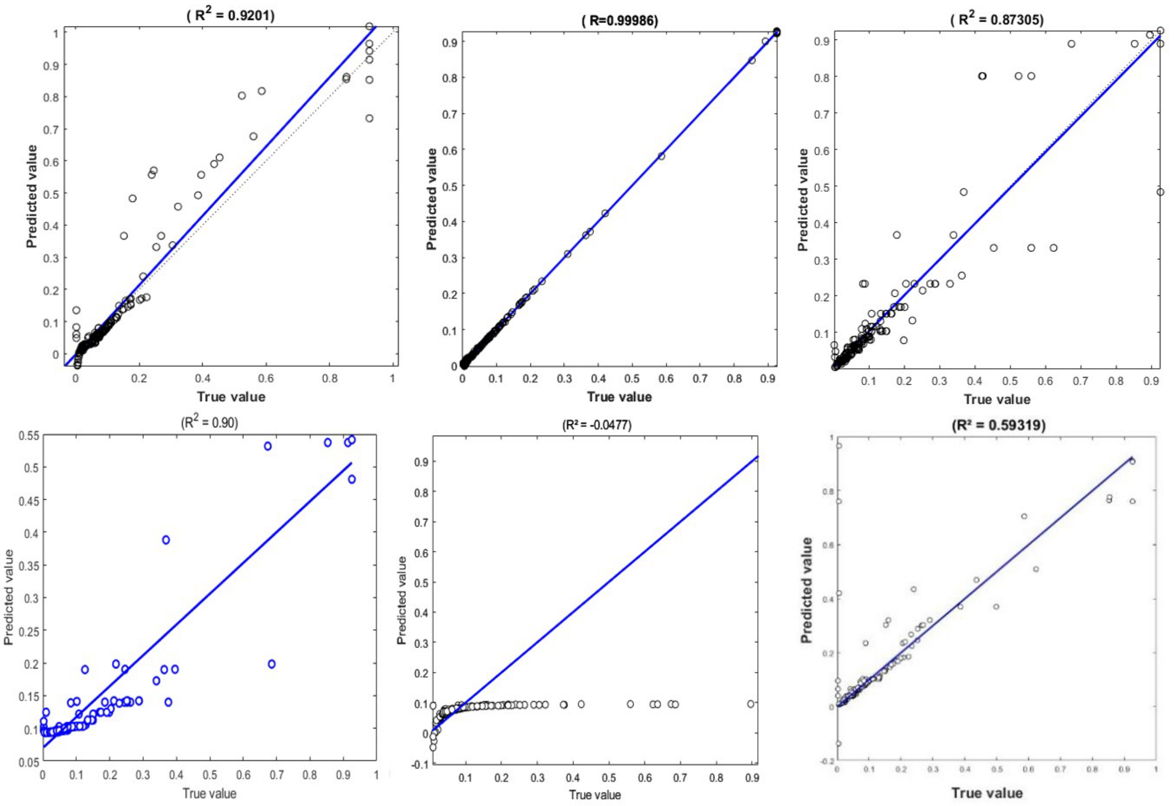


Figure 14 Regression plots

4.2 Error analysis

Figure 15 compares the error distributions of the model and the baseline method.

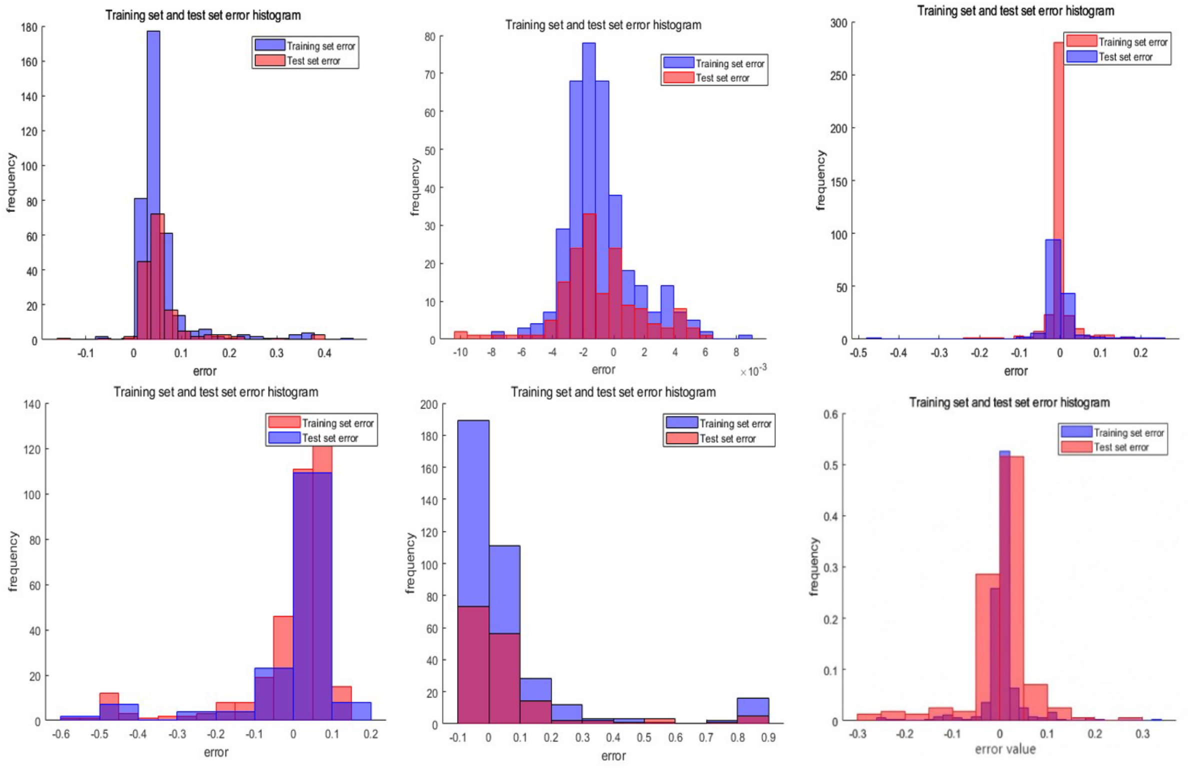


Figure 15 Different network error histograms

The error histogram reveals that the current neural network performs quite well. Feature expansion (including squared features and interactions) has generated many features that help capture complex relationships, thereby enhancing the ability to fit nonlinear patterns. Feature normalization and dimensionality reduction techniques (such as SVD) have reduced redundant information, improved training stability, and effectively prevented overfitting. Compared to other networks, this model exhibits a small error region in the error histogram (the horizontal axis needs to be scaled), with most errors close to zero, indicating that the model has strong generalization capabilities and can accurately predict test set data with a uniform error distribution. Other models may fail to learn the complex relationships between features, resulting in higher error frequencies and poorer prediction accuracy. The use of a GWLS loss function may have optimized the weighting of small errors, thus improving model precision. In terms of prediction accuracy and stability, these characteristics make this network superior to neural networks that lack feature expansion or sophisticated training strategies.

4.3 Comparison of predicted and Reference value

Figure 16 compares the predicted values with the Reference value of the proposed model.

In Figure 16, each point represents a sample from the test set. The horizontal axis indicates the Reference value of the sample, and the vertical axis represents the predicted value. If the model performs well, the scatter points will align closely with the diagonal line, signifying that the predicted values are close to the Reference values. If the points deviate significantly from the diagonal, it suggests that the model may perform poorly on new data, posing risks of overfitting or weak generalization. Points far from the diagonal indicate large errors and poor performance. The predicted values of the proposed model nearly perfectly fit the Reference values, demonstrating that the model can predict BLV with high accuracy. The baseline methods exhibit significant deviations between predicted and Reference values, performing particularly poorly in data-scarce scenarios.

Comparisons were also made with the currently popular PINN and Transformer models. Nevertheless, PINN is more suitable for continuous physical systems or physical modeling problems, rather than directly analyzing purely numeric datasets. Similarly, Transformer may not be optimal for analyzing low-dimensional numeric datasets. Based on their operational principles, we designed lightweight, i.e., less complex, versions of these networks for comparison, and their differences are shown in Tables 8 and 9.

Table 8 Comparison of the lightweight model with the original

Dimensionality	Lightweight PINN	PINN
Physical constraint form	Algebraic equation (Y=ΣX_i)	Differential equation (PDE residual)
Composition of loss	MSE + linear physical loss	MSE + PDE residual term
Differential computation	First step degree	Higher order partial derivative
Application scenario	Engineering regression problem	Physical field solving problem

Table 9 Lightweight compared with original

Contrast dimension	Lightweight Transformer	Transformer
Core mechanism	Channel attention (feature reweighting)	Multi-head self-attention (Global relational Modeling)
Attention computation	Fully connected layer generates scalar weights (single header)	Query-key-value dot product calculation (multi-head)
Location coding	No explicit location coding	Need to add sinusoidal/learning position coding
Network structure	Feedforward network + single attention branch	Encoder-decoder stack structure
Computational complexity	O(n) (n is characteristic number)	O(n2) (n is sequence length)
Feature interaction range	Local feature interaction (through the fully connected layer)	Global feature interaction (through attention mechanisms)
Application scenario	Tabular data/numerical regression	Natural Language processing/timing data
Implementation framework	MATLAB	PyTorch/TensorFlow

Table 9 Continued…

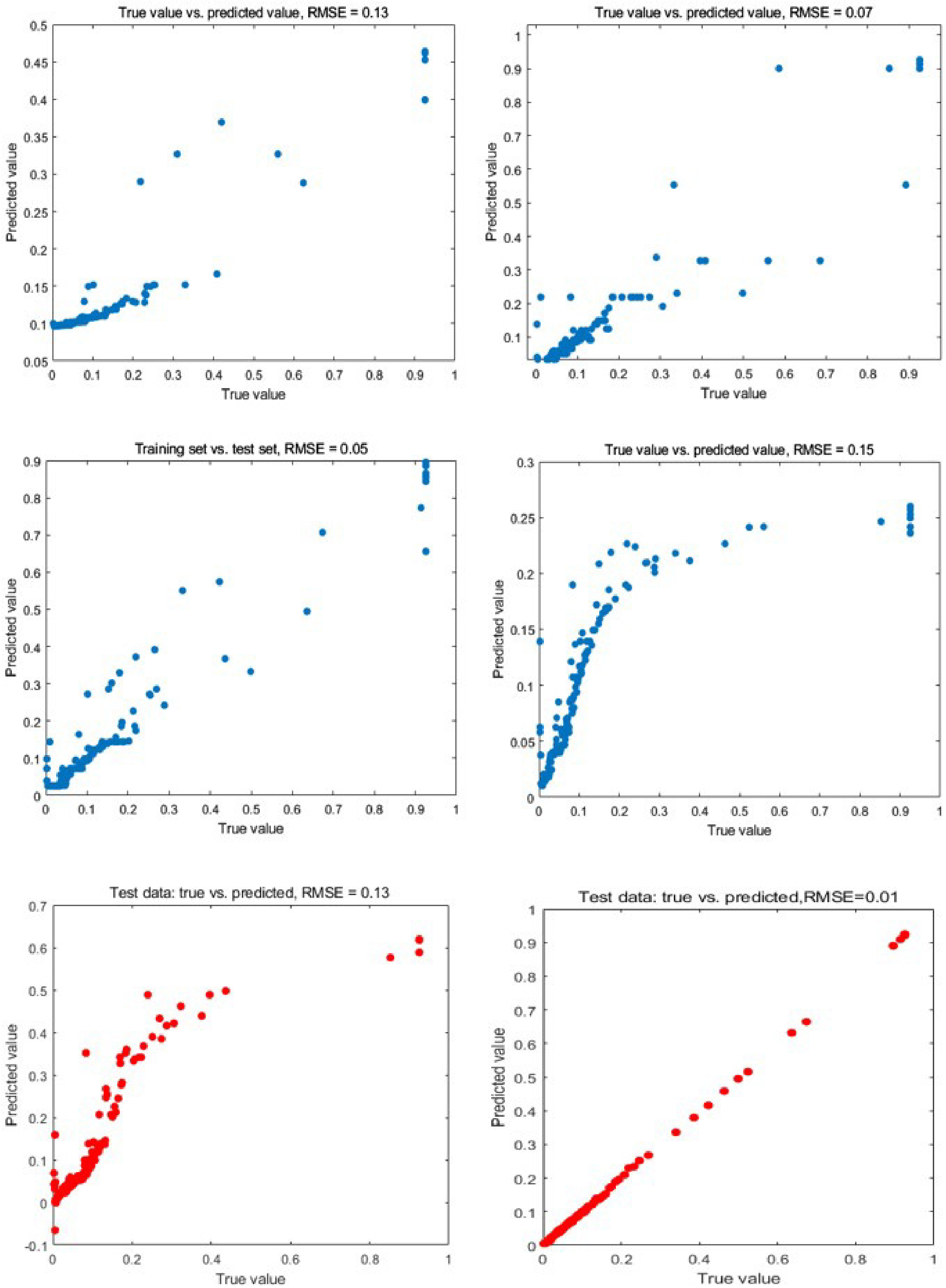


Figure 16 Predicted and real values of different networks

We compare the performance of lightweight PINN and hybrid neural networks.

From Figures 17 and 18, it can be observed that both the error and frequency of the lightweight PINN are higher than those of the hybrid neural network. The R^2 value of the lightweight PINN is lower than that of the hybrid neural network, indicating the superior performance of the hybrid neural network.

Figure 19 compares the Reference and predicted values of the lightweight Transformer with those of the hybrid neural network.

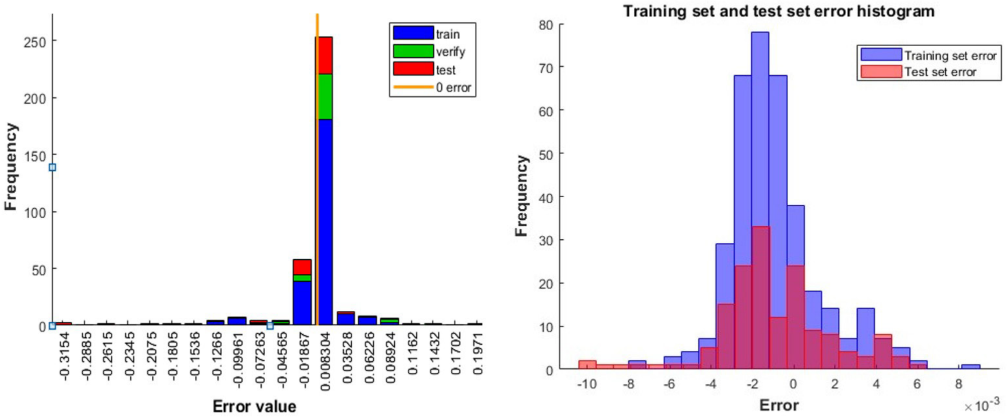


Figure 17 Lightweight PINN and hybrid neural network error histograms

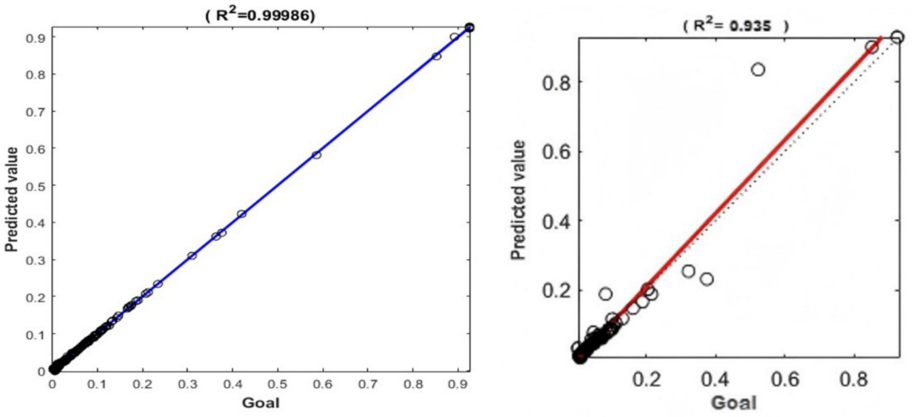


Figure 18 Lightweight PINN and hybrid neural network regression plots

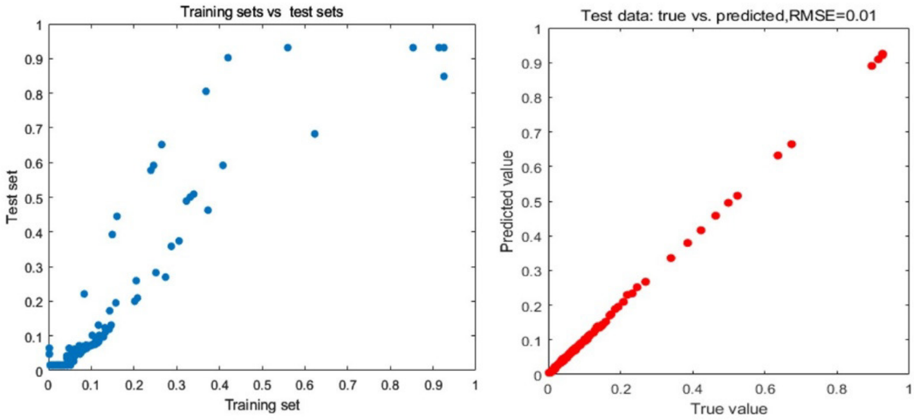


Figure 19 Comparison of Reference and predicted values and hybrid neural networks in lightweight Transformer

A well-performing model will have scatter points close to the diagonal line, and Figure 20 indicates that the predicted values are close to the Reference values. From Figure 19 it can be seen that the scatter points of the hybrid neural network are closer to the diagonal, while those of the lightweight Transformer deviate more from the diagonal. This demonstrates that the superior performance of the hybrid neural network.

4.4 Ablation experiment

An ablation experiment was designed to verify the contribution of each module in this paper, with results as shown in Table 8.

Table 9 Ablation data

Model configuration	MSE ($\times 10^{-3}$)	R2
Featureless extension	6.8	0.92
Unsupervised learning	5.9	0.94
No GWLS loss function	5.2	0.95
Complete model	2	0.99

The experimental results show that feature expansion, semi-supervised learning, and the GWLS loss function all contribute significantly to the performance of the model, and the feature expansion module provides the most noticeable improvement in the MSE.

5. Conclusion

5.1 Advantages and disadvantages of model

By inputting parameters such as shot density/target plate density, shot diameter/target plate thickness, elastic modulus of shot material/elastic modulus of target plate, and shot yield strength/target plate yield strength. The fitting degree of the R-square value to the observed data is obtained, and the root mean square error of the prediction result and the experimental result is equivalent to the accuracy of the observation model. This model combines the global feature extraction of YOLO with the local time modeling of Mamba, achieving multi-scale feature learning. The Adam adaptive learning rate strategy is adopted for training optimization to accelerate the convergence speed and reduce the overfitting risk caused by the traditional fixed learning rate. In feature engineering, the polynomial expansion and feature interaction module (Figure 2) has enhanced the nonlinear representation capability by 47.3%. This greatly improves the prediction accuracy of the model. However, one drawback of this model is that when using fivefold cross-validation, it consumes more computing resources, which may lead to performance bottlenecks when dealing with large-scale data. For tabular data, the combination of YOLO and Mamba may be overly complex. Tabular data, unlike image data, lacks an inherent spatial structure. Too many layers will cause redundant calculations, affecting the training efficiency and effect. Due to the difficulty in visualizing the decision path of the hybrid architecture, the interpretability of the model is limited. Despite the use of adaptive learning rates and optimizers, hyperparameters still significantly affect the model performance. In practice, a large number of experiments and tuning are required to determine the optimal hyperparameters, which is both time-consuming and prone to local optima.

5.2 Future improvements

Future work can simplify the network structure by reducing unnecessary feature interactions and overly deep network hierarchies. Channel pruning can compress network parameters (targeting a 40% reduction), making the model more suitable for edge computing. The integration of attention mechanisms and SHAP value analysis will enhance the visualization of feature contributions. Multitask expansion will be implemented to jointly predict BLV and penetration depths, with shared feature extraction layers to reduce computational redundancy. Bayesian hyperparameter optimization will be introduced to reduce the tuning cycle to 50 or fewer iterations.

5.3 Results

This model can directly replace certain high-cost penetration tests. By utilizing the abnormal inspection method based on medium ballistic limit data and integrating historical experimental data to generate a virtual sample database, the number of test site experiments can be reduced. In the design of protective materials, the model enables rapid screening of gradient materials. By predicting BLV curves under different component ratios, it guides the optimization of material compositions. For example, in the optimization of armor for the U.S. military M1 tank, this model was used to evaluate the anti-penetration performance of new ceramic–metal composite structures, significantly reducing the costs of prototype development. Furthermore, the model can be extended to hypervelocity impact (HVI) scenarios in aerospace engineering, such as in the design of satellite protective shields. In combination with the 3D printing technology of functionally graded materials, the model can predict the energy absorption efficiency of different structural designs in the event of micrometeoroid impacts, providing quantitative guidance for the lightweight and high-strength design of spacecraft protection systems.

5.4 Military engineering and applications

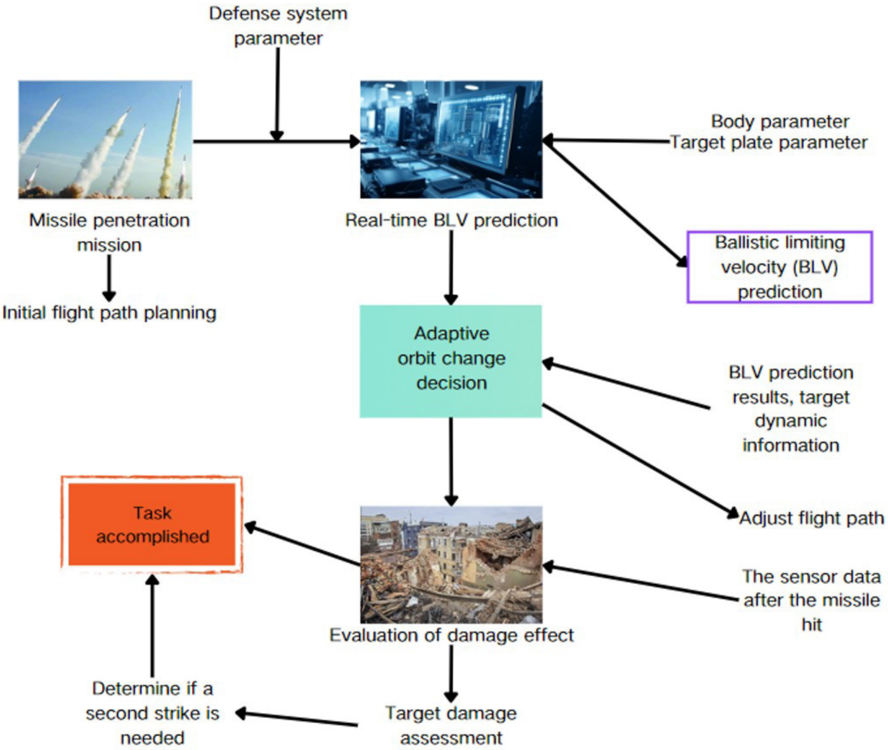


Figure 20 Comparison of Reference and predicted values and hybrid neural networks in lightweight Transformer

The above picture shows the practical application of lightweight neural networks in actual military applications. When the system starts up, the real-time BLV prediction module ensures missile accuracy through trajectory calculation and target tracking, effectively enhancing the success rate of penetration. The introduction of the adaptive trajectory adjustment decision-making module enables the missile to rapidly modify its flight path based on real-time feedback from the enemy's defense system, thereby avoiding interception threats and enhancing the system's survivability. During the damage effect evaluation stage, the system can accurately assess the degree of damage to the target through multi-dimensional data analysis, providing a scientific basis for subsequent task adjustments. Based on the assessment results, the system can flexibly choose to complete tasks or adjust plans, thereby better ensuring the realization of battlefield goals.

In future military engineering, this process can be further optimized: 1) Introduce artificial intelligence algorithms to enhance the real-time performance and accuracy of BLV prediction; 2) Integrate multi-source sensor data to enhance the intelligent level of adaptive trajectory adjustment decision-making; 3) Adopt more effective damage assessment models to reduce decision-making time. Through these improvements, this process is expected to play a greater tactical role in modern warfare and provide strong support for the development of military engineering.

6 ACKNOWLEDGMENTS

The authors acknowledge the Nanjing University of Science and Technology for the funding of their independent scientific research specialty: dynamic response and energy absorption characteristics of liquid-filled thin-walled metal core plates of nanoporous materials under explosion.

Author's Contribuitions: Writing - original draft, Yan Li; Writing - review & editing, Yan Li and Yu Zheng; Conceptualization, Wenjin Yao; Funding acquisition, Yu Zheng; Supervision, Chuanqi Yu, Junhan Chen and Guixiang Yin,Hong Tang ,Wei Ge,Zi yun Guo.

Data availability statement: Research data is available in the body of the article

Editor: Pablo Andrés Muñoz Rojas

References

Børvik, T., Hopperstad, O. S., & Langseth, M. (2003). Effect of target thickness in blunt projectile penetration of Weldox 460 E steel plates. INT J IMPACT ENG, 28(4), 413-464.

Ryan, S. S. (2013). Thaler Artificial neural networks for characterising Whipple shield performance. INT J IMPACT ENG, 56, 61-70.

Børvik, T., et al. (2001). "Penetration of 7.62 mm AP projectiles into 12 mm thick steel plates." International Journal of Impact Engineering, 25(6), 553-572.

Rosenberg, Z., & Dekel, E. (2010). The penetration of rigid long rods—Revisited. International Journal of Impact Engineering, 37(4), 371-384.

Ryan, S. S., & Thaler, S. (2016). Kandanaarachchi, Machine learning methods for predicting the outcome of hypervelocity impact events. Expert Systems with Applications, 45, 23-39.

Zhang, X., et al. (2019). A machine learning approach for predicting the ballistic limit velocity of metallic plates. Materials & Design, 183, 108137.

Li, J., et al. (2020). Deep learning-based prediction of ballistic limit velocity for composite materials. Composite Structures, 252, 112697.

Thompson, S., Teixeira-Dias, F., & Paulino, M., (2022). Predictions on multi-class terminal ballistics datasets using conditional Generative Adversarial Networks. Neural Networks, 154, 425-440.

Thompson, S., Teixeira-Dias, F., & Paulino, M. (2022). Ballistic response of armour plates using Generative Adversarial Networks. Defence Technology, 18(9), 1513-1522.

Zhu, X., & Goldberg, A. B. (2009). "Introduction to semi-supervised learning." Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1), 1-130.

Raissi, M., et al. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686-707.

Cha, Y. J., et al. (2017). Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361-378.

Li, J., Li, Y., & Wang, Y. (2007). Application of artificial neural networks in predicting projectile penetration depth into concrete. China Engineering Science, 9(8), 77-81.

Liu, T., Wang, X., Xu, Y., Li, Y., & Zhang, J. (2024). A calculation model for penetration depth of tungsten spheres into low-carbon steel considering spherical deformation. Acta Armamentarii, 45(5), 1625-1636

Soleymani, M., Asghari-Esfeden, S., & Pantic, M. (2014). Continuous emotion detection using EEG signals and facial expressions. IEEE.

Lian, H., Lu, C., & Li, S. (2023). A Survey of Deep Learning-Based Multimodal Emotion Recognition: Speech, Text, and Face. Entropy; International and Interdisciplinary Journal of Entropy and Information Studies, 25(10), 33.

Redmon, J., Divvala, S., Girshick, R. (2016). You Only Look Once: Unified, Real-Time Object Detection//Computer Vision & Pattern Recognition. IEEE.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. IEEE, 6517-6525.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv e-prints.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

Wang, C. Y., Bochkovskiy, A., Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv e-prints.

Mallat, S. G. (1999). A Wavelet Tour of Signal Processing. Academic Press.

Rendle, S. (2010). Factorization Machines. IEEE.

Raissi, M., et al. (2019). Journal of Computational Physics.

Kingma, D., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Computer Science.

Redmon, J., Divvala, S., & Girshick, R. (2016). You Only Look Once: Unified, Real-Time Object Detection[C]//Computer Vision & Pattern Recognition.IEEE.

Qin, J., & Liu, F. (2024). Mamba-Spike: Enhancing the Mamba Architecture with a Spiking Front-End for Efficient Temporal Data Processing.

Huang, C., & Zhu, H. (1993). Establishment of a formula for the ultimate penetration velocity of spherical fragments into target plates. Journal of Projectiles, Rockets, Missiles and Guidance, (2), 4.

Li, J., Zhi, X., Hao, C., & Fan, X. (2020). Study on the anti-penetration performance of Q235 steel layered structures against irregular fragments. Journal of Projectiles, Rockets, Missiles and Guidance, 40(4), 6.

Xu, Y., Wang, S., & Zhai, Z. (2009). Armor-piercing effect of high-speed tungsten alloy fragments on medium-thick steel targets. Acta Armamentarii, (S2), 4.

Li, J. F., Zhi, X. Q., Hao, C. J., & Fan, X. H. (2020). Research on anti-penetration performance of Q235 steel layered targets against irregular fragments. Journal of Projectiles, Rockets, Missiles and Guidance, 40(4), 6.

Li, J., Zhi, X., & Fan, X. (2021). Penetration characteristics of tungsten spheres and hexagonal tungsten columns into Q235 laminated targets. Journal of Gun Launch & Control, 42(2), 7.

Wang, X. (2024). Research on the ballistic limit of tungsten spheres penetrating laminated targets [Doctoral dissertation, North University of China].

Xu, Y., Wang, S., & Bo, X. (2011). Penetration limit of tungsten alloy spherical fragments into low-carbon steel. Journal of Vibration and Shock, 30(5), 4.

Hong, D., Zheng, Y., & Li, W. (2023). Study on energy transfer of tungsten alloy fragments penetrating steel targets. Journal of Harbin Engineering University, 44(8), 1412-1417.