


RESEARCH

Open Access



Improved Meta-learning Neural Network for the Prediction of the Historical Reinforced Concrete Bond–Slip Model Using Few Test Specimens

Chengwen Zhang¹ , Qing Chun^{1*}, Ao Sun², Yijie Lin¹ and HaoYu Wang¹

Abstract

The bond–slip model plays an important role in the structural analysis of reinforced concrete structures. However, many factors affect the bond–slip behavior, which means that a large number of tests are required to establish an accurate bond–slip model. This paper aims to establish a data-driven method for the prediction of the bond–slip model of historical reinforced concrete with few test specimens and many features. Therefore, a new Mahalano-bis-Meta-learning Net algorithm was proposed, which can be used to solve the implicit regression problem in few-shot learning. Compared with the existing algorithms, the Mahalanobis-Meta-learning Net achieves fast convergence, accurate prediction and good generalization without performing a large number of tests. The algorithm was applied to the prediction task of the bond–slip model of square rebar-reinforced concrete. First, the first large pretraining database for the bond–slip model, BondSlipNet, was established containing 558 samples from the existing literature. The BondSlipNet database can be used to provide a priori knowledge for learning. Then, another database, named SRRC-Net, was obtained by 16 groups of pull-out tests with square rebar. The SRRC-Net database can be used to provide the posteriori knowledge. Finally, based on the databases, the algorithm not only successfully predicted the bond–slip model of square rebar-reinforced concrete, but also that of the other 23 types of reinforced concrete. The research results can provide a scientific basis for the conservation of square rebar-reinforced concrete structures and can contribute to the bond–slip model prediction of the other types of reinforced concrete structures.

Keywords: neural network, bond–slip model, meta-learning, architectural heritage, square rebar

1 Introduction

The accuracy of the bond–slip constitutive model between the rebar and concrete influences the reliability of the calculation results of the bearing capacity (Nabilah et al., 2020; Wang et al., 2021), regardless of whether theoretical derivation or finite element numerical simulation is applied. However, many new types of concrete or rebar

still result in new bond–slip behavior, which needs to be studied. For example, square section rebar and concrete with a special mix ratio were found in historical buildings in China (built in approximately 1910s–1950s). This type of square rebar-reinforced concrete is abbreviated as SRRC hereafter. As shown in Fig. 1, SRRC concrete has unique proportions and different material configuration requirements. At the same time, SRRC rebar was mainly ribbed square rebar, which is very different from the current deformed rebar, thereby determining a unique bond–slip model (Zhang et al., 2021a, 2021b). Therefore, the prediction of the bond–slip model between square rebar and SRRC concrete is a new prediction task.

Journal information: ISSN 1976-0485 / eISSN 2234-1315

*Correspondence: cqj1979@163.com

¹ School of Architecture, Southeast University, Nanjing 210096, China
Full list of author information is available at the end of the article

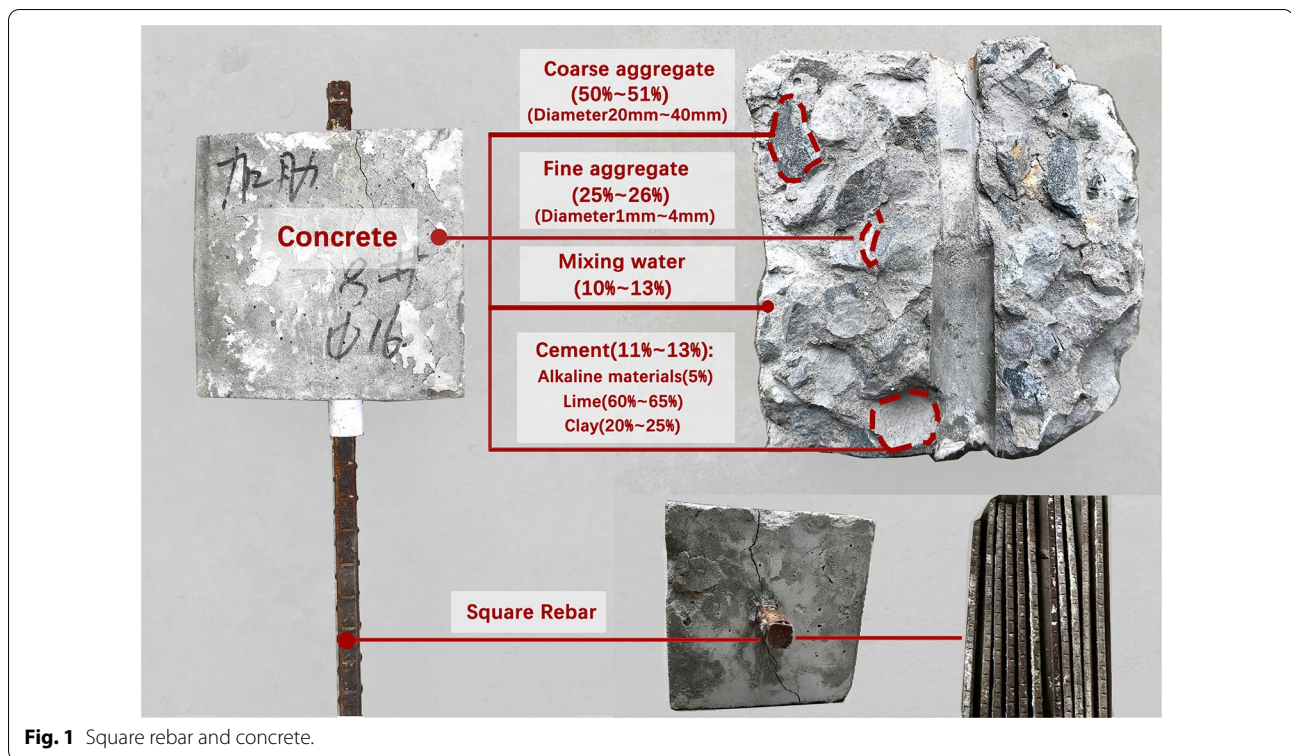


Fig. 1 Square rebar and concrete.

In recent years, many meaningful studies have been performed on the bond–slip problem. The smooth surface caused by epoxy coating or zinc coating will reduce the bonding performance. In addition, sand-coating treatment on the rebar surfaces will improve bonding performance (Islam et al., 2020; Pauletta et al., 2020; Rabi et al., 2020; Wang et al., 2020; Zhou & Qiao, 2018). Some studies show that the types of concrete, such as green concrete, high-strength concrete, fine aggregate concrete, and recycled aggregate concrete, also exert a certain effect on the bond–slip performance (Liu et al., 2020; Nguyen et al., 2020; Paswan et al., 2020; Wang, 2016; Wardeh et al., 2017). According to the literature (Fu et al., 2021; Hou et al., 2018; Yang et al., 2015; Zhou et al., 2020a, 2020b), the corrosion of rebar exerts a very complex effect on the bond–slip properties. On the one hand, the iron rust attached to the surface of the rebar will increase the friction between the rebar and the concrete, but on the other hand, the uneven volume expansion will cause the concrete near the rebar to crack, reducing the grip force between the concrete and the rebar. In addition, the rib shape, spacing and projection area of rebar (Cai et al., 2020; Li et al., 2019; Metelli & Plizzari, 2014; Prince & Singh, 2015), the cover thickness (Rockson et al., 2020), the diameter and spacing of the stirrups (Koulouris & Apostolopoulos, 2020), the

size of specimens (Zhang et al., 2020a, 2020b), and the loading modes (pull-out, beam-type drawing or beam-type bending) (Kaffetzakis & Papanicolaou, 2016; Rockson et al., 2020) all affect the bond–slip model.

Thus, the bond–slip model between rebar and concrete is very complex, and scholars have made valid attempts to determine the bond–slip model. Based on the elastic mechanical solution for a thick-walled cylinder under constant pressure, the analytical solution was derived for the key point coordinates of the bond–slip curve (Gao et al., 2019). However, as shown in Fig. 1, the SRRC rebar has a square section, which does not validate the hypothesis in the derivation process, and the results of SRRC rebar contain complex variable functions, which are not convenient for engineering applications. The application of two-dimensional and three-dimensional finite element methods for bond–slip models was proposed (Biscaia & Soares, 2020; Liu et al., 2020). However, the premise of finite element calculation depends on accurate prior knowledge of the bond–slip behavior. Since the friction coefficient and chemical interactions between rebar and concrete are difficult to accurately test, it is challenging to use the finite element method to directly predict the bond–slip model. A unified formula was constructed for a bond–slip model by collecting a large amount of test data in the literature (Wu & Zhao, 2013). However,

the present results only have a good confidence level for the bond–slip model between ordinary concrete and deformed rebar. At the same time, the model expression (polynomial, exponential expression, or logarithmic expression) of bond–slip needs to be determined according to the experience of researchers. Therefore, if the prediction task changes, the regression parameters and regression function expressions in the new and old tasks may be difficult to balance.

Therefore, data-driven method can be used to solve the complex problem of prediction (Cai et al., 2019; Jang et al., 2019; Kakavand et al., 2021; Vanneschi et al., 2018). A deep neural network (DNN) method was proposed to predict the bond strength between fiber reinforced polymers (FRPs) and concrete, thereby obtaining good results (Naderpour et al., 2019; Zhou et al., 2020a, 2020b). However, the bond–slip relationship between rebar and concrete is a new research topic. Moreover, only the ultimate bond strength was predicted, and the other characteristic points of the rising stage, descent stage and residual stage were not predicted, which means that the prediction task in this paper contains more features of the output. In fact, the existing neural network technology for predicting bond strength often uses the traditional DNN framework. The results of this training method depend on the characteristic distribution of the dataset and the size of the dataset. It is difficult to use the existing network for transfer learning with a small amount of test data for new types of concrete or rebar. In contrast, a large number of tests must be performed to study the strength of concrete, specimen size, stirrup arrangement, cover thickness, rib shape, proportion of concrete and other factors and then train a new DNN for learning, which consumes considerable resources and time. In addition, a large number of square rebar samples are difficult to obtain for tests to build a large neural network training database because SRRC reinforced concrete buildings are mainly cultural relic protection units. Fortunately, a large number of tests have been conducted on bond–slip between other types of rebar and concrete, which provide a considerable degree of prior knowledge. Moreover, the bond behavior of the steel–concrete composite structure is very complex and has similar characteristics to rebar-reinforced concrete (Wang et al., 2019), which contains a sufficient number of features for training and provides prior knowledge for the prediction task. Although different prediction tasks will lead to differences in the final bond–slip model, some basic knowledge about the model can be shared, which provides the basis for the Mahalanobis-Meta-learning Net (MMN).

In summary, to solve the task of using a small amount of test data to predict the bond–slip model, such as

SRRC concrete reinforced with square rebar, the following method was proposed in this paper. First, in Sect. 2, the databases used for training were established. Based on the test results from 36 studies, a database named BondSlipNet was established by determining the main factors affecting the bond–slip model between rebar and concrete. This database can be used to provide prior knowledge through the so-called pretraining process. Then, the pull-out test of the SRRC reinforced concrete was designed to obtain the bond–slip curve data, which can be used to establish the database for the SRRC task named SRRC-Net. This database is different from the BondSlipNet database due to its few samples, and it is used to provide posterior knowledge through the so-called fine-tuning process. Second, in Sect. 3, based on the improvement in the model-agnostic meta-learning (MAML) algorithm (Finn et al., 2017), the MMN algorithm was established and used to solve the prediction task of the SRRC bond–slip model. It should be noted that although the prediction task of the SRRC bond–slip model was used as an example, the MMN algorithm and the BondSlipNet database can be used to solve other prediction tasks with few test specimens and many features, which is discussed in Sect. 4. Finally, the MMN algorithm was compared with other algorithms to verify the prediction ability and generalization level of the network.

2 Dataset Preparation

2.1 BondSlipNet Database for Prior Knowledge

In this paper, based on 36 references (Biscaia & Soares, 2020; Cai et al., 2020; Coccia et al., 2016; Fu et al., 2021; Hou et al., 2018; Islam et al., 2020; Kaffetzakis & Papanicolaou, 2016; Khaksefidi et al., 2021; Koulouris & Apostolopoulos, 2020; Leibovich et al., 2020; Li et al., 2019, 2020; Liu et al., 2020; Metelli & Plizzari, 2014; Mo & Chan, 1996; Nguyen et al., 2020; Paswan et al., 2020; Pautletta et al., 2020; Prince & Singh, 2013, 2015; Qi et al., 2020; Rabi et al., 2020; Rafi, 2019; Rockson et al., 2020; Wang, 2016; Wang et al., 2018, 2020; Wardeh et al., 2017; Xiao & Falkner, 2007; Xing et al., 2015; Yang et al., 2015; Yeih et al., 1997; Zhang et al., 2020a, 2020b, 2021a, 2021b; Zhou & Qiao, 2018; Zhou et al., 2020a, 2020b) with 2039 experimental tests, a large bond–slip database containing 558 samples (after averaging repeated groups), BondSlipNet, is labeled and established. The slip value was measured using an extensometer, and the bond stress was determined by rebar strain gauge or drawing force. Limited by the manuscript length, 2000 tests were collected from 36 articles, and after considering the repetitive samples, 558 samples were ultimately included. These samples mainly focus on the pull-out test and also cover a small amount of beam pull-out and bending tests.

In fact, many other meaningful research results that can provide prior knowledge, such as lap splice tests, are still available. These valuable results remain to be added to the dataset in future studies. However, the main innovation of this paper is the method used to construct large databases and the task-based MMN algorithm based on existing databases. Algorithms and database construction methods can be easily applied to new data for test results, such as FRP rod-reinforced concrete and lap splice tests.

The following assumptions were used to simplify the labeling process:

(1) If the labeled parameters were not specified in the original literature, they could be determined according to other relevant literature (Béton 1990; Gao et al., 2019; CAoB 2010).

(2) Since the coefficient of friction between rebar and concrete is difficult to be determined quantitatively, only qualitative sorting was used according to the surface roughness to distinguish sand-coating surfaces, rusted surfaces, ordinary surfaces, epoxy-coated surfaces, zinc-coated surfaces, etc., during the labeling process.

(3) If the concrete batching contained substitute materials, the substitute materials were removed from the proportion of concrete, which means that only the proportion of ordinary batching was calculated to distinguish concrete with special batching.

(4) If the rebar yields or the concrete breaks, the descending stage is not tracked in most literature. Therefore, it is assumed that the bond stress reaches 0 after a very small amount of relative slip, and at this time, the remaining characteristic points of the descending stage and the residual stage are linearly interpolated.

(5) Since the Combin 39 element used to simulate the bond–slip characteristics in ANSYS only supports 20 feature points, the output curve feature points in this database are selected as 20. Ten of the feature points with a uniform abscissa distribution were used to describe the behavior of the ascent stage, and the other ten feature points were used for the descent stage.

(6) Since task-based MMN learning is performed in the following section, BondslipNet distinguishes different prediction tasks based on five subitems. The subitems include different concrete types, rebar types, specimen cross-sectional shapes, loading modes, and stirrup arrangement modes. When dividing the samples into different tasks, there may be some samples that belong to more than two tasks. In this case, these samples will be divided equally to ensure that each task will not have too few samples.

Then, the samples in BondSlipNet were divided into 23 prediction tasks. Among these 23 prediction tasks, 2 tasks will be selected randomly in Sect. 4 to construct the test set together with the SRRC tasks, which will not be fed to the network until the test process. More details about the method for dividing the training set and test set will be discussed in Sect. 4. In the following text, the above different classifications are abbreviated in Table 1. For example, specimen #6 was labeled $A_{cir}B_{pla}C_{nor}D_{nsti}E_{nor}$, which means that the specimen is normal concrete with a circular section and normal rebar without stirrups.

The BondSlipNet database contains three sets of data: data X (features), data $Y-S$ (slip), and data $Y-T$ (bond stress). Among them, X , as the resource of the input layer of the subsequent network, contains a total of 558 samples, and each sample has 24 features, as shown in Table 2 (24 features are from the row of Water to the row of Coefficient of friction). $Y-S$ and $Y-T$ are the resources of the output layers of the subsequent network, each containing 558 samples (namely, curves of bond–slip) corresponding to X . In general, each sample of the output layer contains 40 features (20 points for bond stress $Y-T$ and 20 points for slip $Y-S$), which are the point coordinates in the curves. It should be noted since the training process of the DNN is based on the samples and that of the MMN is based on the tasks, the resource of the input/output layers means the training/test set was established

Table 1 Letter abbreviation used in the prediction task classification.

Section of concrete specimen		Loading modes			Concrete types			
Square	Circular	Beam-type bending	Beam-type drawing	Pull-out	Coarse aggregate replaced	Fine aggregate replaced	Water replaced	Normal concrete
A_{squ}	A_{cir}	B_{bwan}	B_{bla}	B_{pla}	C_{arep}	C_{srep}	C_{nwat}	C_{nor}
Stirrups arrangement mode			Rebar types					
With stirrups		Without stirrups		Corroded rebar	Plain rebar	Coated rebar	Normal rebar	
D_{sti}		D_{nsti}		E_{cor}	E_{pla}	E_{coa}	E_{nor}	

Table 2 Data *X* in the BondSlipNet database.

Serial number	Parameter range	#1	#2	#3	...	#558
Water	0~2.04	0.58	0.58	0.58	...	0.83
Cement	0~1	1.00	1.00	1.00	...	1.00
Sand	0~3.33	2.13	2.13	2.13	...	2.52
Aggregate	0~6.12	2.93	2.93	2.93	...	4.82
Fc (MPa)	21.64~102.62	35.90	35.90	35.90	...	44.60
Compressive strength						
Da (mm)	0~31	20.00	20.00	20.00	...	20.00
Maximum aggregate size						
fy (MPa)	206~1374	410.00	410.00	410.00	...	422.00
Rebar yield strength						
c (mm)	20~225	45.25	45.25	45.25	...	63.00
Concrete cover thickness						
ds (mm)	6~50	9.50	9.50	9.50	...	14.00
Rebar diameter						
As (mm ²)	28.27~1963.5	70.88	70.88	70.88	...	153.94
Rebar area						
S (mm)	0~240	0.00	0.00	0.00	...	0.00
Stirrup spacing						
dsv (mm)	0~10	0.00	0.00	0.00	...	0.00
Stirrup diameter						
Es (MPa)	157,575~257,536	203,000	203,000	203,000	...	209,000
Rebar elastic modulus						
Ec (MPa)	1523~44,240	31,580	31,580	31,580	...	33,579
Concrete elastic modulus						
Esv (MPa)	0~210,000	0.00	0.00	0.00	...	0.00
Stirrup elastic modulus						
L (mm)	15~370	40.00	60.00	80.00	...	70.00
Embedded length						
Sr (mm)	0~15	5.92	5.92	5.92	...	9.00
Rib spacing						
Lr (mm)	0~3	1.33	1.33	1.33	...	1.06
Rib top length						
Lb (mm)	0~4.8	2.11	2.11	2.11	...	3.14
Rib bottom length						
Hr (mm)	0~1.82	0.52	0.52	0.52	...	1.40
Rib height						
Alpha (°)	0~90	65.00	65.00	65.00	...	53.20
Rib angle						
Degree of decay (%)	0~13.5	0.00	0.00	0.00	...	0.00
Ac (mm ²)	3632~250,000	7854	7854	7854	...	19,600.00
Concrete section area						
Coefficient of friction	-1~1.5	1.00	1.00	1.00	...	1.00
Number of test pieces	/	18			...	36
Task type	/	A _{cir} B _{pla} C _{nor} D _{nsti} E _{nor}	A _{cir} B _{pla} C _{nor} D _{nsti} E _{nor}	A _{cir} B _{pla} C _{nor} D _{nsti} E _{nor}	...	A _{squ} B _{pla} C _{nor} D _{nsti} E _{nor}
Reference	/	Yeih et al., (1997)			...	Qi et al., (2020)

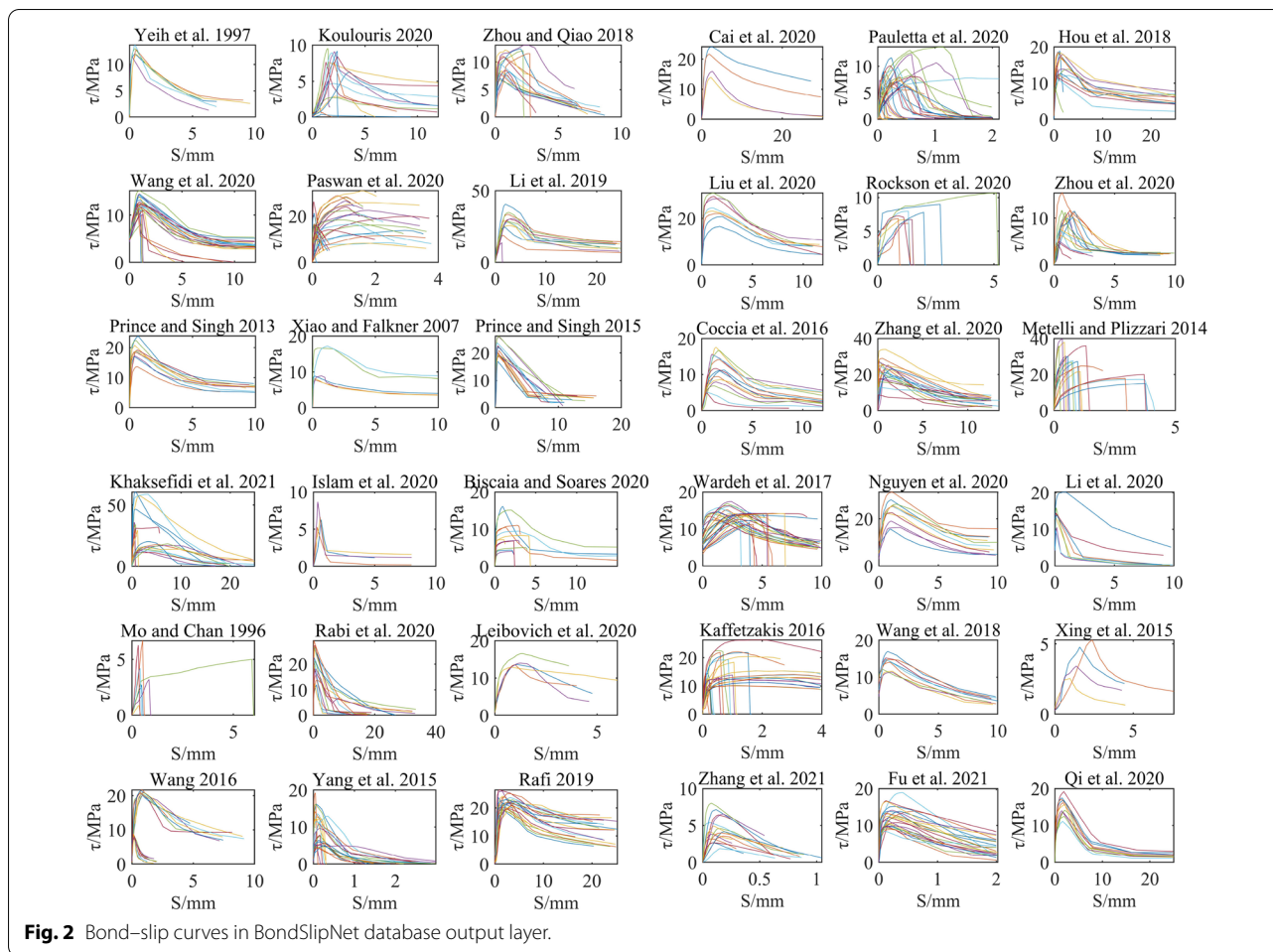
Due to the large number of samples, this table is only an example. More details can be found in the Additional file 1, which can be published with this paper online. The water, cement, sand and aggregate are the relative values, assuming that the value of cement is 1

indirectly or directly based on the resource. *Y-S* and *Y-T* are plotted as shown in Fig. 2.

2.2 SRRC-Net Database for Posterior Knowledge

Although the BondSlipNet database has been established, the tests described below were designed using

SRRC square rebar to obtain the distribution characteristics of the SRRC reinforced concrete bond–slip model (Zhang et al., 2021a, 2021b). Sixteen groups of pull-out specimens were prepared considering 16 different conditions, such as different batches of concrete, different rebar section sizes, different cover thicknesses, different



stirrup ratios, and different degrees of corrosion. Each group contains duplicate test specimens. The 16 groups of test details were labeled according to the format of data X in BondSlipNet, as an example shown in Table 3. A 1000 kN hydraulic testing machine was used as the device to apply the pull-out force. The test specimens and loading device are shown in Fig. 3.

At displacement test points 1 and 2, the displacement relative to the fixed end (S_a , S_b) is measured using a laser displacement meter. At displacement test point 3, the displacement of the fixed end (S_c) is measured using a laser extensometer. The pull-out force P is measured by a load sensor on the machine and applied by the oil pressure control method at a speed of 2 mm/min. The bond–slip curves of the 16 groups of specimens are shown in Fig. 4, and the key point details can be found in the Additional file 2.

Figs. 2 and Fig. 4 show that since the section, rib form and concrete ingredients of the SRRC specimens were special, the bond strength, peak slip value, characteristic point coordinates and overall trend of the final bond–slip curves were different from those of modern reinforced concrete.

Now, the databases (BondSlipNet and SRRC-Net) for the MMN algorithm in this paper have been basically completed. The specific algorithm of the MMN will be introduced in the next section.

3 Training Network Construction

3.1 MAML Model Algorithm

For many years, the training sample size has been an important issue that cannot be avoided when using neural networks for prediction. The prediction ability of a neural network based on a database with a small

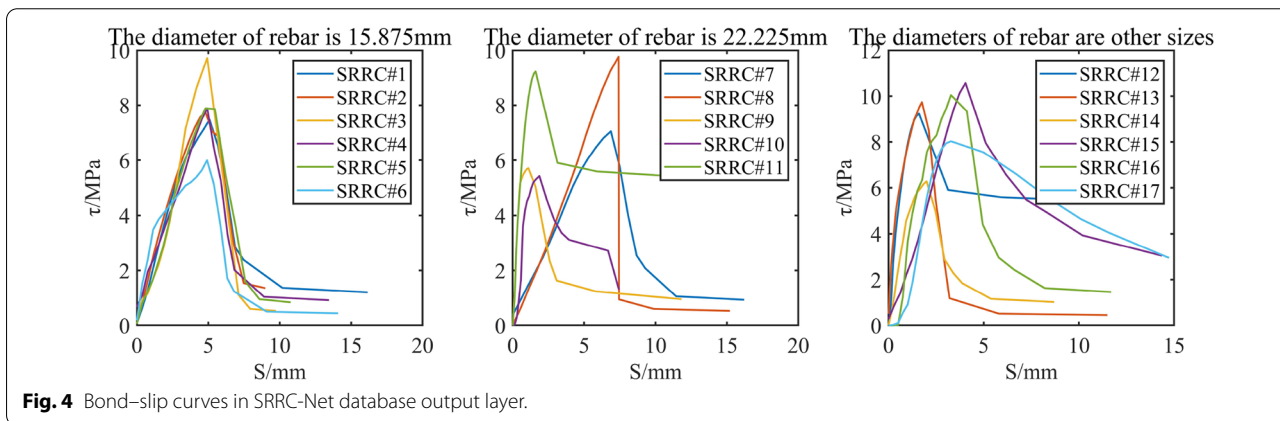
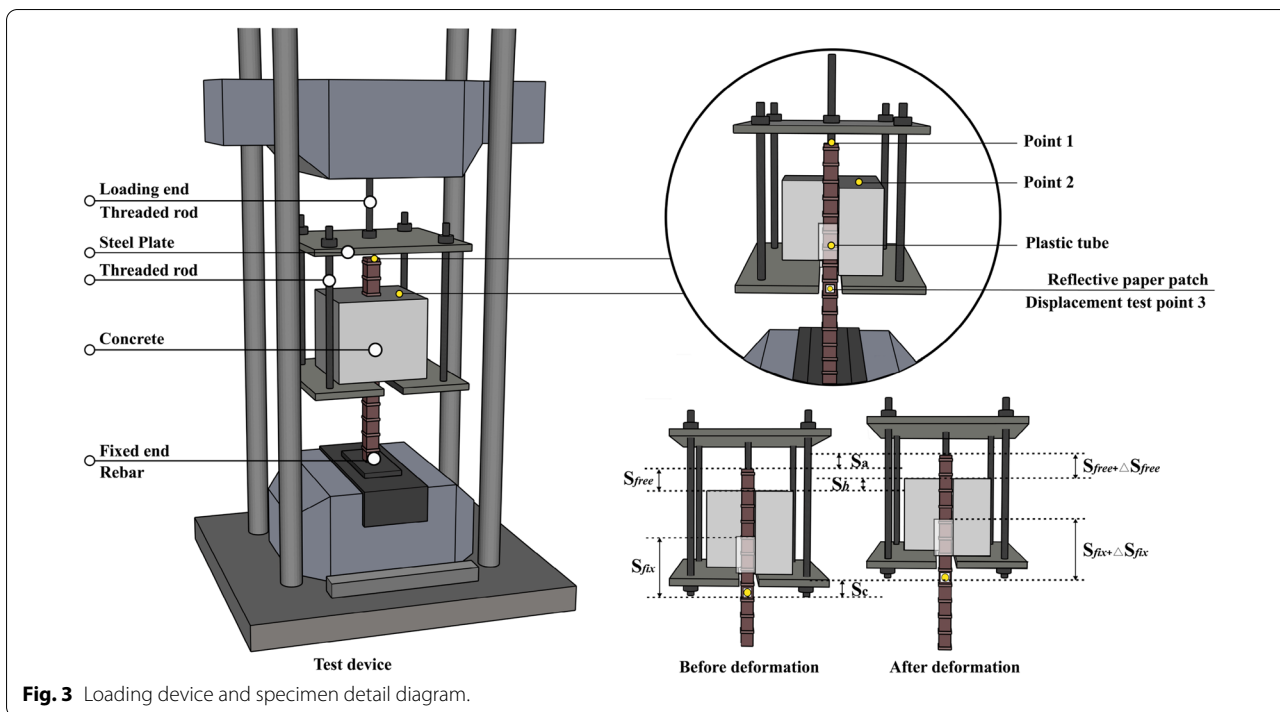
Table 3 Data *X* in the SRRC-Net database.

Serial number	SRRC#1	SRRC#2	SRRC#3	SRRC#4	SRRC#5	SRRC#6	...	SRRC#16
Water	0.68	0.68	0.68	0.68	0.68	0.68	...	0.7
Cement	1	1	1	1	1	1	...	1
Sand	1.68	1.68	1.68	1.68	1.68	1.68	...	1.73
Aggregate	3.4	3.4	3.4	3.4	3.4	3.4	...	3.6
F _c (MPa) Compressive strength	14.94	14.94	14.94	14.94	14.94	14.94	...	15.22
D _a (mm) Maximum aggregate size	30	30	30	30	30	30	...	25
F _y (MPa) Rebar yield strength	278.6	278.6	278.6	278.6	278.6	278.6	...	218.0
C (mm) Concrete cover thickness	71.44	71.44	71.44	71.44	71.44	71.44	...	99.00
D _s (mm) Rebar diameter	15.88	15.88	15.88	15.88	15.88	15.88	...	19.50
A _s (mm ²) Rebar area	252.02	252.02	252.02	252.02	252.02	252.02	...	374.00
S (mm) Stirrup spacing	0	0	0	0	0	0	...	0
D _{sv} (mm) Stirrup diameter	0	0	0	0	0	0	...	0
E _s (MPa) Rebar elastic modulus	181,670	181,670	181,670	181,670	181,670	181,670	...	123,035
E _c (MPa) Concrete elastic modulus	14,100	14,100	14,100	14,100	14,100	14,100	...	13,360
E _{sv} (MPa) Stirrup elastic modulus	0	0	0	0	0	0	...	0
L (mm) Embedded length	79.38	79.38	79.38	79.38	79.38	79.38	...	110.00
S _r (mm) Rib spacing	16	11	16	16	16	16	...	20
L _r (mm) Rib top length	1	1	1	1	1	1	...	10
L _b (mm) Rib bottom length	1.2	1.2	1.2	1.2	1.2	1.2	...	15.0
H _r (mm) Rib height	1.5	1.5	1.5	1.5	1.5	1.5	...	2.0
Alpha (°) Rib angle	0	0	0	0	0	0	...	0
Degree of decay (%)	0	0.10	0.97	0.86	0.26	0.68	...	0
A _c (mm ²) Concrete section area	25,600	25,600	25,600	25,600	25,600	25,600	...	48,400
Coefficient of friction	1.00	1.10	1.25	1.25	1.15	1.25	...	1.00

Due to the large number of samples, this table is only an example. More details can be found in the Additional file 2, which can be published with this paper online

sample size is unable to easily reach an ideal accuracy. However, in the big data era, there is almost no unique task in most of the existing prediction tasks that cannot find any similar tasks to provide certain prior knowledge. Taking the bond–slip model prediction of SRRC reinforced concrete as an example, although only a

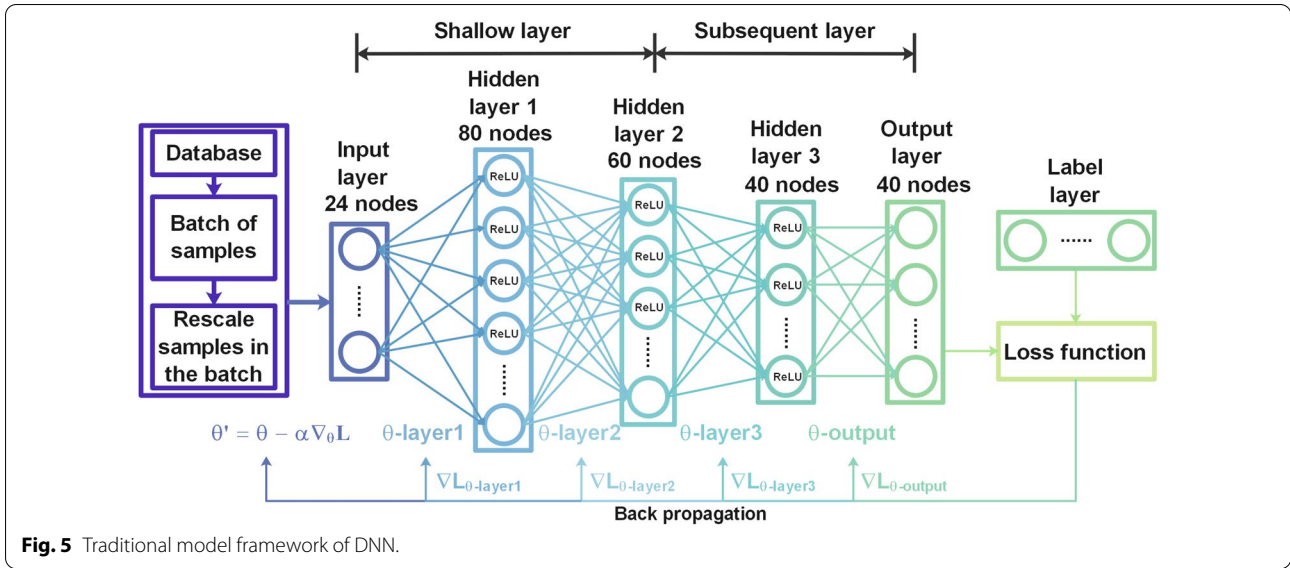
few tests have been performed before, the bond–slip performance between other types of rebar and concrete has been extensively studied. In fact, the function expression form of the bond–slip curve or the range of parameters of the expression can be easily changed with various influencing factors. However, some properties



have common characteristics, such as the stronger the tensile strength of concrete is, the larger the maximum bond stress will be under the premise of no rebar yield and rib failure. Alternatively, the existence of stirrups will improve the ductility of the concrete and increase the residual bond stress. These common characteristics provide a good basis of knowledge for neural network learning. Therefore, when the sample size of the target prediction task is small, it is recommended to conduct

prior knowledge learning on a large dataset with similar task objectives and then transfer it to the target task. In the field of deep learning, this problem is called few-shot learning, and the MAML algorithm has been proposed in the literature (Finn et al., 2017) to solve this problem.

The traditional DNN model is shown in Fig. 5, which can be described as follows and can be realized by Box 1.



Box 1 Algorithm for the traditional DNN

Input: Point set X with 24 characteristics and 558 samples
 Label set Y with 40 characteristics and 558 samples
Training parameter: θ -weights and bias in the net
Hyper parameter: epoch-how many times each sample was trained
 α -parameters to control the learning rate of net

Step:

1. Divide the samples into training set and test set
2. θ initialization
3. **for** $i=1, \dots, \text{epoch}$
4. **for** batches of samples in the training set
5. Get $\hat{y}(X, \theta)$ with Forward propagation
6. Calculate the loss function $L = \sum_{i=1}^m (y - \hat{y}(X, \theta))^2 / m$
7. Update θ to $\theta' = \theta - \alpha \nabla_{\theta} L(X, Y, \theta)$
8. **end**
9. **end**

Output: θ

Notation "i" indicates the current training times, "m" refers to the sample number, $\hat{y}(X, \theta)$ represents the predicted value, "L" represents the loss function of the regression problem, θ' refers to the updated weights and biases in the current training times, "y" represents the labeled value, and ∇ is the gradient of the variable.

For a certain distribution $P(X, Y, \theta)$, X is the input sample, and Y is the output result. The dimensions of X are $m \times n_x$, where m is the number of samples and n_x is the number of features of each sample. The dimensions of Y are $m \times n_y$, where n_y is the number of features of each output, and θ is the parameters of the neural network (containing weight, bias and other learning parameters). For each different task, θ starts from random initialization, and then the predicted value $\hat{y}(X, \theta)$ of each iteration step will be obtained after forward propagation. Generally, let the loss function of the regression problem be $L = \frac{1}{m} \sum_{i=1}^m (y - \hat{y}(X, \theta))^2$ (measuring the error between the predicted value and the labeled value). Finally, the parameters are updated by calculating $\theta' = \theta - \alpha \nabla_{\theta} L(X, Y, \theta)$ until the most suitable θ is found to describe the corresponding relationship between X and Y , where ∇_{θ} represents the gradient vector to θ , and α represents the learning rate. However, in many engineering problems, the gradient descent process of neural networks is not the optimization process of smooth convex functions. If the number of samples is small and the distribution of new and old tasks is different, the network will have difficulty escaping from the saddle point or prematurely entering the local optimum, thereby obtaining lower prediction accuracy.

The traditional DNN training objects are the samples, namely, X itself. The goal of the MAML algorithm is to ‘learn how to learn’, and the training objects are tasks, that is, θ . In other words, the DNN framework training object is each sample point in BondSlipNet, and the MAML framework training object is first the tasks divided from BondSlipNet, and then is the sample in the specific prediction task. Therefore, the MAML algorithm can be divided into two parts. The first part is meta-learning, which uses large databases to train with different tasks and finally obtains θ^* with good generalization performance. Then, θ^* is used for a small sample database, and after a small amount of gradient update, the final model is obtained, which is the second part called the fine-tuning process.

The MAML algorithm is shown in Fig. 6, which can be described as follows and can be realized as Box 2.

The global task can be divided into B batches, and one batch will be extracted for updating each time. Assuming that the number of tasks in a batch is m_B , the global model parameter is initialized to θ , and the training set R and test set R' are extracted from the m_t th task. First, the training set R is trained, and the model parameter is updated by the gradient. Then, for the R' set, the loss is calculated, and the model parameter is updated. The loss function of the m_t th task is set to l_{m_t} , and the parameters for m_B training

tasks in this batch are initialized to θ . The parameters of the m_t th task will be changed to $\theta_{m_t}^i$ using iterative Eq. (1) after i times of updates:

$$\theta_{m_t}^i = \theta_{m_t}^{i-1} - \alpha \nabla_{\theta_{m_t}^{i-1}} l_{m_t}(\theta_{m_t}^{i-1}), \tag{1}$$

α is the learning rate for each task.

After the training tasks in each batch are established, the loss function of the global model is set to $L(\theta)$, then:

$$L(\theta) = \sum_{m_t=1}^{m_B} l_{m_t}(\theta_{m_t}^i) = \sum_{m_t=1}^{m_B} l_{m_t}(\theta_{m_t}^{i-1} - \alpha \nabla_{\theta_{m_t}^{i-1}} l_{m_t}(\theta_{m_t}^{i-1})). \tag{2}$$

$L(\theta)$ is the functional of θ , where m_B represents the number of tasks processed in each batch. The global parameter θ can be updated to θ' using Eq. (3):

$$\begin{aligned} \theta' &= \theta - \beta \nabla_{\theta} L(\theta) = \theta - \beta \nabla_{\theta} \sum_{m_t=1}^{m_B} l_{m_t}(\theta_{m_t}^i) \\ &= \theta - \beta \sum_{m_t=1}^{m_B} \nabla_{\theta} l_{m_t}(\theta_{m_t}^i). \end{aligned} \tag{3}$$

Note that both $\theta(w_j), j = 1, 2, 3, \dots, k$ and $\theta_{m_t}^i(w_{l,m_t}^i), l = 1, 2, 3, \dots, k$ contain k parameters, and the second item of Eq. (3) is written as Eq. (4):

$$\nabla_{\theta} l_{m_t}(\theta_{m_t}^i) = \begin{bmatrix} \partial l_{m_t}(\theta_{m_t}^i) / \partial w_1 \\ \vdots \\ \partial l_{m_t}(\theta_{m_t}^i) / \partial w_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^k \frac{\partial l_{m_t}(\theta_{m_t}^i)}{\partial w_{l,m_t}^i} \frac{\partial w_{l,m_t}^i}{\partial w_1} \\ \vdots \\ \sum_{l=1}^k \frac{\partial l_{m_t}(\theta_{m_t}^i)}{\partial w_{l,m_t}^i} \frac{\partial w_{l,m_t}^i}{\partial w_j} \\ \vdots \end{bmatrix}, \tag{4}$$

w is the component of array θ , which refers to a single value of weight or bias. Similarly, Eq. (1) can be rewritten as Eq. (5):

$$\begin{aligned} w_{l,m_t}^1 &= w_l - \alpha \frac{\partial (l_{m_t}(\theta))}{\partial w_l} \\ &\vdots \\ w_{l,m_t}^i &= w_{l,m_t}^{i-1} - \alpha \frac{\partial (l_{m_t}(\theta_{m_t}^{i-1}))}{\partial w_{l,m_t}^{i-1}} \\ &\vdots \end{aligned} \tag{5}$$

Combining Eqs. (4), (5), $\frac{\partial w_{l,m_t}^i}{\partial w_j} = \frac{\partial w_l}{\partial w_j} - \alpha \left(\frac{\partial^2 l_{m_t}(\theta)}{\partial w_l \partial w_j} + \sum_{s=1}^{i-1} \frac{\partial^2 l_{m_t}(\theta_{m_t}^s)}{\partial w_{l,m_t}^s \partial w_j} \right)$ can be obtained. Then, each component w_j^i of $\theta^i(w_j^i), j = 1, 2, 3, \dots, k$ in Eq. (3) can be simplified as Eq. (6):

Box 2 Algorithm for traditional MAML

Input: Point set X with 24 characteristics and 558 samples
 Label set Y with 40 characteristics and 558 samples

Training parameter: θ -weights and bias in the net

Hyper parameter: **epoch1**-how many times each task was trained in the MAML
epoch2-how many times each task was trained in the Fine-tune
 α - parameters to control the learning rate of net
 β - parameters to control the learning rate of task
 i - parameters to control the update times in each task

Steps:

1. Divide the samples into training set and test set
2. Divide the training set into M tasks, the test set into N tasks

Steps in the MAML:

1. θ initialization
2. **for** $j = 1, \dots, \text{epoch1}$
3. **for** batches of tasks in M tasks
4. θ_{temp} initialization
5. Get $\hat{y}(X, \theta)$ with Forward propagation
6. Calculate the loss function of each task $l_{m_i} = \sum_{i=1}^m (y - \hat{y}(X, \theta))^2 / m$
7. Update θ_{temp} for i times to get $\theta_{m_i}^i = \theta_{m_i}^{i-1} - \alpha \nabla_{\partial \theta_{m_i}^{i-1}} l_{m_i}(\theta_{m_i}^{i-1})$
8. **end**
9. Get the loss function of each batch $L(\theta) = \sum_{m_i=1}^{m_B} l_{m_i}(\theta_{m_i}^i) = \sum_{m_i=1}^{m_B} l_{m_i}(\theta_{m_i}^{i-1} - \alpha \nabla_{\partial \theta_{m_i}^{i-1}} l_{m_i}(\theta_{m_i}^{i-1}))$
10. Update w'_j in the θ by $w'_j = w_j - \beta \sum_{m_i=1}^{m_B} \partial l_{m_i}(\theta_{m_i}^i) / \partial w_{i,m_i}^j$
11. **end**

Output: $\theta'(w'_j)$

Steps in the Fine-tune for a special task in N tasks:

1. Froze the w'_j in the shallow layer of θ'
2. **for** $j = 1, \dots, \text{epoch2}$
3. Repeat the steps 4-8 in DNN(Box 1)
4. **end**

Output: θ'

Notation "j" refers to the current training times, "m" represents the sample number, $\hat{y}(X, \theta)$ is the predicted value, θ represents the updated weights and biases in the current training times, "y" is the labeled value, θ_{temp} is a temporary parameter used to search for gradient descent directions, m_B represents the number of tasks in a batch, m_i represents the m_i -th task in the train, ∂ represents a partial differential notation, and w is the component of array θ , which represents a single value of weight or bias.

$$w'_j = w_j - \beta \sum_{m_t=1}^{m_B} \sum_{l=1}^k \frac{\partial l_{m_t}(\theta_{m_t}^i)}{\partial w_{l-m_t}^i} \left(\frac{\partial w_l}{\partial w_j} - \alpha \left(\frac{\partial^2 l_{m_t}(\theta)}{\partial w_l \partial w_j} + \sum_{s=1}^{i-1} \frac{\partial^2 l_{m_t}(\theta_{m_t}^s)}{\partial w_{l-m_t}^s \partial w_j} \right) \right), \quad (6)$$

It should be noted that the MAML has made two simplifications in deriving Eq. (6), that is, only one gradient update is conducted in meta-learning, and the role of the Hessian is ignored. Through mathematical derivation and computational experimental analysis, it can be concluded that for implicit learning tasks, these two simplifications may reduce the accuracy of the training network, and the specific explanations will be given in Sect. 3.3.2 and Sect. 4.3.

At this point, the training of one batch in the MAML algorithm is completed, and then the aforementioned process is repeated to train and update the remaining batches until all batches are input into the network to complete the training, which is regarded as the completion of one epoch. It should be noted that in each epoch, the selection of task sets is random, which has a similar effect as cross-validation. After several epochs, the meta-learning process is completed. Based on the obtained optimal initialization parameters, traditional DNN training is performed here for the specific tasks using a database with a small sample size. Then, the parameters of the shallow layer in the DNN are frozen, and only the parameters in the subsequent layer are trained, after which the fine-tuning process is finally completed. The shallow layer represents the first several layers, and the subsequent layer represents the last several layers. In this study, the parameters in the shallow layer indicate those between the Input layer, the Hidden layer 1 and Hidden layer 2. The parameters in the subsequent layer indicate those between the Hidden layer 2, Hidden layer 3 and Output layer.

3.2 Modification of the MAML–MMN

The original MAML algorithm was used to train the regression problem with the sine function as an example, which obtained good results. However, for the vast majority of engineering problems, an explicit relationship is often unavailable between the input and output of data. The training of the network is not only aimed at obtaining the regression of several parameters but also needs to determine the number of parameters and even the function itself or the implicit relationship between the data. At the same time, many regression problems in engineering have larger input feature dimensions, output feature dimensions and smaller sample sizes (compared with those of image classification problems). Therefore, many problems arise when directly using

the MAML algorithm to predict the bond–slip model of reinforced concrete (more details are provided in Sect. 4).

In this paper, the MMN algorithm is created according to the particularity of the implicit regression problem of the bond–slip model, as shown in Fig. 7.

Compared with the MAML algorithm, the following improvements are incorporated in the MMN algorithm.

(1) For the expression of the loss function, the single-layer perceptron is used to modify the Mahalanobis distance loss, replacing the mean square error (MSE) loss.

(2) Multiple gradient updating is considered in meta-learning.

(3) The overall framework is changed into a multitask learning framework. The output task is divided into two tasks, namely, the prediction of the slip stage curve and the prediction of the failure stage curve, which use joint learning (Sun et al., 2020). The multitask learning framework plays a dimension reduction role in learning tasks. Furthermore, when using a multitask learning framework for joint training, the different tasks establish linkages between the minimum value through shared parameter constraints. Thus, the multitask learning framework has a parameter sharing mechanism, which improves the generalization of the network and reduces the risk of overfitting. Dropout (Srivastava et al., 2014) and L2 regularization technology (Rahaman et al., 2018) were added to improve the generalization level of the network, and gradient clip (Zhang et al., 2020a, 2020b) was added to avoid gradient explosion. BN normalization is changed to FRN normalization, which also prevent the model from overfitting (Singh & Krishnan, 2020).

The MMN algorithm can be realized using the Box 3. Mathematical derivation will be performed in Sect. 3.3 to analyze its improvement significance. Among them, the improvements in (3) have already been explained by many studies. Therefore, the next section mainly focuses on the explanation of improvement points (1)–(2).

3.3 Mathematical Explanation of the MMN

3.3.1 Modified Mahalanobis Distance Loss

Let the output results of the MMN equal $\hat{y}_{a \times b}$, where \mathbf{a} represents the sample size and \mathbf{b} represents the output feature dimension. Assuming that the number of predicted feature points in this task is m , then $\mathbf{b} = 2 \times m$. $\hat{y}_{a \times b}$ is rewritten as matrix $\hat{y}_{2 \times \frac{ab}{2}}$ with 2 rows and $\frac{ab}{2}$

Box 3 Algorithm for MMN

Input: Point set X with 24 characteristics and 558 samples
 Label set Y with 40 characteristics and 558 samples
Training parameter: $\theta 1$ -weighs and bias in the net of slip stage curve
 $\theta 2$ -weighs and bias in the net of failure stage curve
Hyper parameter: **epoch1**-how many times each task was trained in the MAML
epoch2-how many times each task was trained in the Fine-tune
 α_1, α_2 -parameters to control the learning rate of net
 β_1, β_2 -parameters to control the learning rate of task
i - parameters to control the update times in each task

Steps:

1. Divide the samples into training set and test set
2. Divide the training set into M tasks, the test set into N tasks
3. Rescale each characteristic with divided by the maximum values
4. Divide the output task into two parts: slip stage and failure stage curve **Improvement (3)**

Steps in the MAML:

1. $\theta 1$ and $\theta 2$ initialization
3. **for** j = 1,...,epoch1
3. **for** batches of tasks in M tasks
4. $\theta 1_{temp}$ **and** $\theta 2_{temp}$ initialization
5. Get $\hat{y}(X, \theta)$ with Forward propagation
6. Calculate the loss function of each task:

$$l_{m_i}(y_{2 \times \frac{ab}{2}}, \hat{y}_{2 \times \frac{ab}{2}}) = \overline{tr} \left[\begin{pmatrix} \vdots & \vdots \\ \Delta S_i & \Delta \tau_i \\ \vdots & \vdots \end{pmatrix} \begin{bmatrix} c'_{11} & c'_{12} \\ c'_{21} & c'_{22} \end{bmatrix} \begin{pmatrix} \cdots & \Delta S_i & \cdots \\ \cdots & \Delta \tau_i & \cdots \end{pmatrix} \right] \quad \text{Improvement (1)}$$

7. Update $\theta 1_{temp}$ for **i** times to get $\theta^i_{m_i} = \theta^{i-1}_{m_i} - \alpha_1 \nabla_{\theta^{i-1}_{m_i}} l^1_{m_i}(\theta^{i-1}_{m_i})$
9. Update $\theta 2_{temp}$ for **i** times to get $\theta^i_{m_i} = \theta^{i-1}_{m_i} - \alpha_2 \nabla_{\theta^{i-1}_{m_i}} l^2_{m_i}(\theta^{i-1}_{m_i})$
8. **end**
9. Get the loss function of each batch $L(\theta) = \sum_{m_i=1}^{m_g} l^1_{m_i}(\theta^i_{m_i}) + \sum_{m_i=1}^{m_g} l^2_{m_i}(\theta^i_{m_i})$
10. Update w'_j in the $\theta 1$ and $\theta 2$ by Equation (6)

$$w'_j = w_j - \beta \sum_{m_i=1}^{m_g} \sum_{l=1}^k \frac{\partial l_{m_i}(\theta^i_{m_i})}{\partial w^l_{l,m_i}} \left(\frac{\partial w_l}{\partial w_j} - \alpha \left(\frac{\partial^2 l_{m_i}(\theta)}{\partial w_l \partial w_j} + \sum_{s=1}^{i-1} \frac{\partial^2 l_{m_i}(\theta^s)}{\partial w_l \partial w_j} \right) \right) \quad \text{Improvement (2)}$$

11. **end**

Output: $\theta'(w'_j)$

Steps in the Fine-tune for a special task in N tasks:

1. Froze the w'_j in the shallow layer of θ'
2. **for** j = 1,...,epoch2
3. Repeat the steps 4-8 in DNN(Box 1)
4. **end**

Output: θ'

For simplification, Drop out, L2 regularization and FRN normalization were not included in the above box but in the Fig.7. It is easy to obtain the influence caused by these terms. Notation \overline{tr} is defined as an operator for obtaining the mean value after taking diagonal elements. ΔS_i is the difference between the predicted slip value and the labeled slip value. $c'_{11}, c'_{12}, c'_{21}$, and c'_{22} are learnable parameters. The meanings of other notations are the same as those mentioned above.

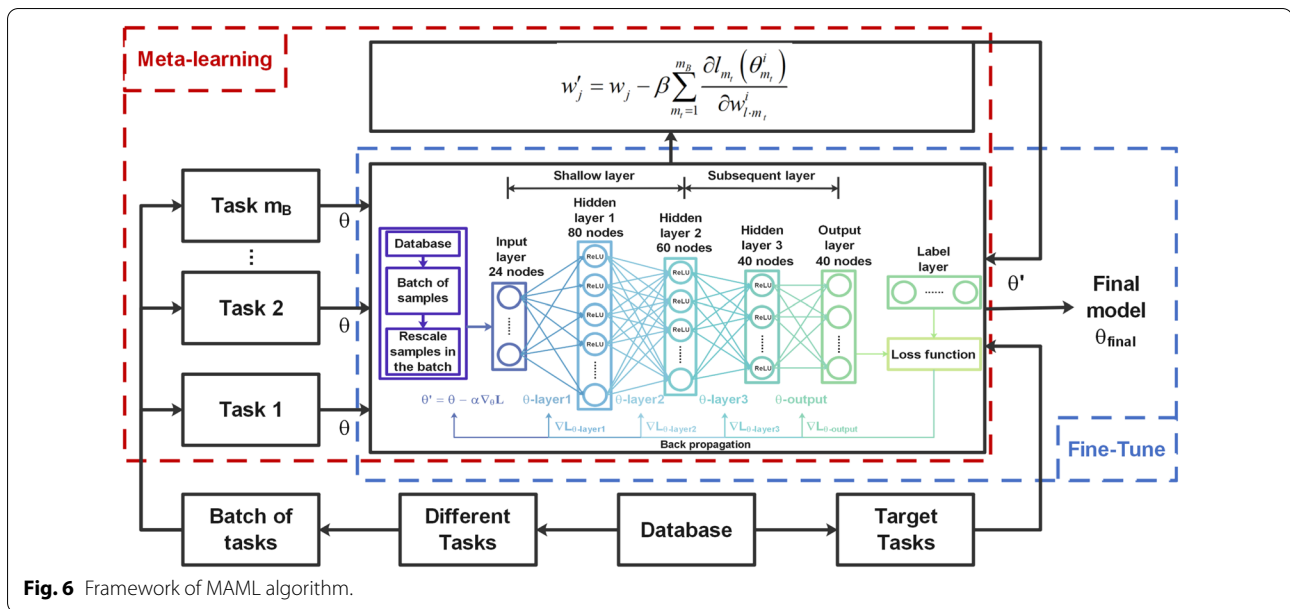


Fig. 6 Framework of MAML algorithm.

columns. Then, the first row of $\hat{y}_{2 \times \frac{ab}{2}}$ is the abscissa (slip S_i) of the predicted points, and the second row is the ordinate (bond stress T_i). Similarly, the labels in the dataset are set as $y_{2 \times \frac{ab}{2}}$.

The Mahalanobis distance loss between the output and the label item can be calculated by Eq. (7):

$$l_{m_i} \left(y_{2 \times \frac{ab}{2}}, \hat{y}_{2 \times \frac{ab}{2}} \right) = \overline{tr} \left[\left(\hat{y}_{2 \times \frac{ab}{2}} - y_{2 \times \frac{ab}{2}} \right)^T C^{-1} \left(\hat{y}_{2 \times \frac{ab}{2}} - y_{2 \times \frac{ab}{2}} \right) \right], \tag{7}$$

\overline{tr} is defined as an operator for finding the mean value after taking diagonal elements and C^{-1} is the inverse matrix of the covariance matrix of matrix $Y_{2 \times ab} = \left[\hat{y}_{2 \times \frac{ab}{2}}, y_{2 \times \frac{ab}{2}} \right]$. Equation (7) shows that the Mahalanobis distance is equivalent to the following progress. First, principal component analysis (PCA) of the data points in the sample space will be performed. Then, the sample space is rotated according to the principal component so that the dimensions are independent of each other. Finally, the distance between the sample points can be obtained by standardization.

Only when the covariance matrix is a unit matrix, that is, when each dimension is independent and identically distributed, the Mahalanobis distance degenerates to the Euclidean distance. Thus, it can be seen that the Euclidean distance treats the relationship between the dimensions of the sample ‘fairly’, ignoring the different distribution characteristics between various dimensions. For the implicit regression problem of the bond–slip model, the mapping between the input features and the

output coordinate points has multiple expressions or can be regarded as an implicit relationship. The horizontal and vertical coordinates of the output coordinate point have different practical significance and exhibit different distribution characteristics. For example, it is assumed that there is such a set of data points as shown in Fig. 8 that satisfy the distribution $P(x, y)$ of the predicted bond–slip model. Among them, the coordinate of point A is (2.5 mm, 5.8 MPa), and the initial output of that point is A_1 (5.5 mm, 6.8 MPa), and the second output is A_2 (3.5 mm, 2.8 MPa), both of which have the same Euclidean distance between points A. However, it is obvious that A_2 is more likely to meet the distribution $P(x, y)$.

Equation (7) shows that the Mahalanobis distance between two points is independent of the measurement unit, which can eliminate the interference of correlation between variables. However, the Mahalanobis distance can be easily affected by outlier samples, thus sacrificing the overall accuracy.

Therefore, the single-layer perceptron was introduced to modify the matrix C^{-1} , which is one type of neural network with only one layer. In essence, to weaken the effect of the outliers, the modified Mahalanobis distance using single-layer perceptron enables the element values of covariance matrix of the original Mahalanobis distance to participate in the training process. Since the number of outliers is generally small, and the covariance matrix of Mahalanobis distance is a 2×2 matrix, the learning ability of single-layer perceptron is sufficient for this task. Therefore, the modified Mahalanobis distance obtained by single-layer perceptron has strong anti-noise ability. At the same time, the complexity of the model

will not be increased substantially due to the small number of parameters introduced by the single-layer perceptron, which is convenient to ensure the training speed and generalization ability. Let $C^{-1} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$, $C'^{-1} = \begin{bmatrix} \eta \tan h(\omega_{11}c_{11} + \delta_{11}) & \eta \tan h(\omega_{12}c_{12} + \delta_{12}) \\ \eta \tan h(\omega_{21}c_{21} + \delta_{21}) & \eta \tan h(\omega_{22}c_{22} + \delta_{22}) \end{bmatrix} = \begin{bmatrix} c'_{11} & c'_{12} \\ c'_{21} & c'_{22} \end{bmatrix}$, where the coefficient η limits the parameter interval, which constrains the effect of the outliers on C^{-1} . The perceptron is initialized by $C^{-1} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$, the learning parameters are ω_{ij} , δ_{ij} ($i, j = 1, 2$) and η , and then C^{-1} is updated to further learn the distribution of data.

The corrected C'^{-1} is substituted into Eq. (7). Equation (8) can be obtained as follows:

$$l_{m_t}(y_{2 \times \frac{ab}{2}}, \hat{y}_{2 \times \frac{ab}{2}}) = \bar{tr} \left[\begin{pmatrix} \vdots \\ \Delta S_i \Delta \tau_i \\ \vdots \end{pmatrix} \begin{bmatrix} c'_{11} & c'_{12} \\ c'_{21} & c'_{22} \end{bmatrix} \begin{pmatrix} \cdots & \Delta S_i & \cdots \\ \cdots & \Delta \tau_i & \cdots \end{pmatrix} \right]. \quad (8)$$

Then, let $W = c'_{11} \Delta S_i^2 + (c'_{12} + c'_{21}) \Delta S_i \Delta \tau_i + c'_{22} \Delta \tau_i^2$. Thus, in addition to weighting the slip deviation and bond stress deviation in the Euclidean distance, the product of bond stress deviation and slip deviation is also considered. Let $S \sim (\bar{S} \pm \sigma(S))$ and $\tau \sim (\bar{\tau} \pm \sigma(\tau))$, if $\sigma(S) \approx \sigma(\tau)$, the data dispersion degree of the two distributions is approximate, and then the role of $\Delta S_i \Delta \tau_i$ can be ignored (and vice versa), which should also validate the above analysis.

Finally, $W = c'_{11} \Delta S_i^2 + (c'_{12} + c'_{21}) \Delta S_i \Delta \tau_i + c'_{22} \Delta \tau_i^2$ can be regarded as a cone curve about ΔS_i and $\Delta \tau_i$, so it must satisfy $W > 0$ to have practical significance. Therefore, the modified Mahalanobis distance loss function proposed in this paper can add the following strengthened constraint condition: $-2 < \frac{(c'_{12} + c'_{21})}{\sqrt{c'_{11} c'_{22}}} < 2$ and $c'_{11} > 0$.

3.3.2 Hessian Matrix

In Sect. 3.1, the parameter updating Eq. (6) in meta-learning is derived. The Hessian matrix term essentially measures the second-order sensitivity of the loss function $l_{m_t}(\theta)$ of each task in the meta-learning stage to the network parameters. The Hessian matrix term reflects the curvature of the global loss function and can help escape saddle points and local minima during gradient descent. In addition, in Sect. 3.3.1, the following conclusion is derived: for the implicit regression problem of the bond-slip model, the loss function $l_{m_t}(\theta)$ is related to $\Delta S_i \Delta \tau_i$, which means that there is a certain internal connection between the neurons in the output layer. In addition to sharing certain parameters between the layers, a second-order effect is observed between the parameters of some

neurons. Therefore, the Hessian matrix term cannot be omitted.

For the explicit regression problem, since the number of parameters to be regressed is known, the network parameter space may search for the direction of global parameters according to the gradient direction of the update results of the internal task gradient of meta-learning, as shown in Fig. 9. However, for implicit regression problems, it is necessary to use the Hessian matrix correction term to search the parameter space of the mapping mode according to the curvature variation characteristics of the loss function, and then the direction of updating the global parameters in this parameter space is searched.

4 Results and Discussion

To verify the generalization level of the MMN prediction method, in addition to the SRRRC bond-slip prediction task, task-A ($A_{\text{squ}} B_{\text{pla}} C_{\text{nor}} D_{\text{nsti}} E_{\text{coa}}$) and task-B ($A_{\text{squ}} B_{\text{pla}} C_{\text{nwat}} D_{\text{sti}} E_{\text{nor}}$) were randomly selected to test the network performance. It should be noted that task-A and task-B are randomly selected and never trained by the network. Therefore, it can be regarded as a test set to verify whether the algorithm can be used in practical engineering applications. In general, the ratio of the training and validation datasets ranges from 7:3 to 8:2. However, when the dataset is small, the trade-off between bias and variance is inevitable. Many scholars, such as Finn et al. (2017), have attempted to solve this problem, and few-shot learning has been proposed. In the problem of few-shot learning, some special operators, such as MAML, have been used to train a better model with only a few training and test samples, which includes the learning framework to weaken the effect of overfitting. Therefore, the samples of each task were divided into a training set and a validation set with a proportion of approximately 10:1 during meta-learning. On this partition ratio, for the SRRRC tasks, task-A and task-B, the training set is only used during fine-tuning, and the test set will only be fed to the MMN when testing the net performance. In the present study, the training and validation datasets were not fixed at each batch or epoch. After completing one batch or epoch of learning, the model will shuffle the dataset and resample at the ratio of 10:1.

At present, the four-stage linear model and the three-stage nonlinear model are mostly used in the prediction models of bond-slip between rebar and concrete. Therefore, in this paper, the test results are compared with the calculation results of the eight models described in Table 4. The models were realized using the python 3.6.2 and tensorflow 1.6.0 in the Jupyter Notebook. The tuning process for hyperparameters can

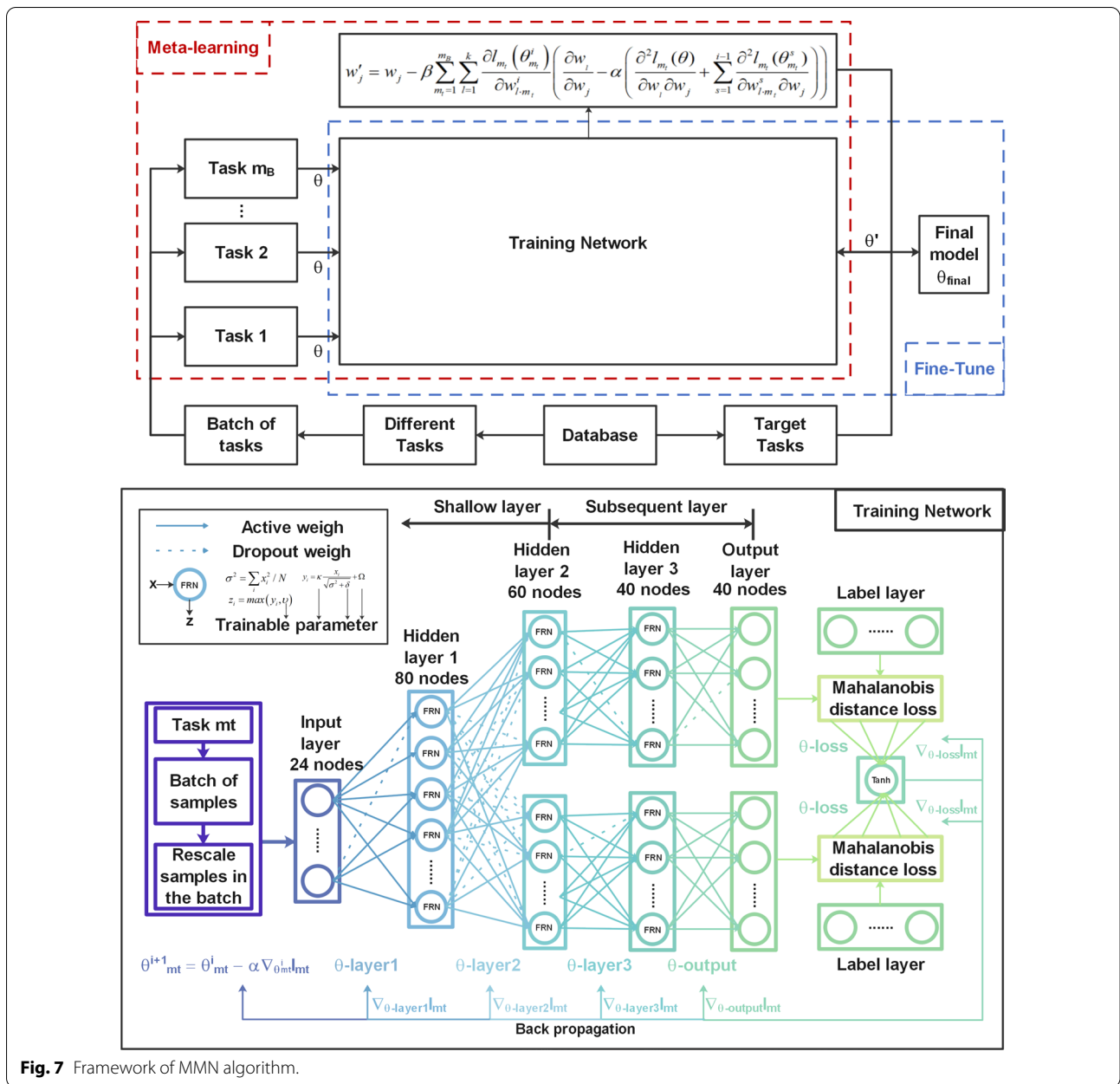


Fig. 7 Framework of MMN algorithm.

be described as follows, which were determined based on the time cost and the accuracy of the results. The epoch in Box 1 and epoch1 or epoch2 in Boxes 2 and 3 were changed from 30 to 60, and finally set to 50. The learning rates, such as α and β , were changed from 0.01 to 0.00001, and finally took 0.0001. The hyperparameters \mathbf{i} , which were used to control the update times in each task, were changed from 2–10, and finally set to 6.

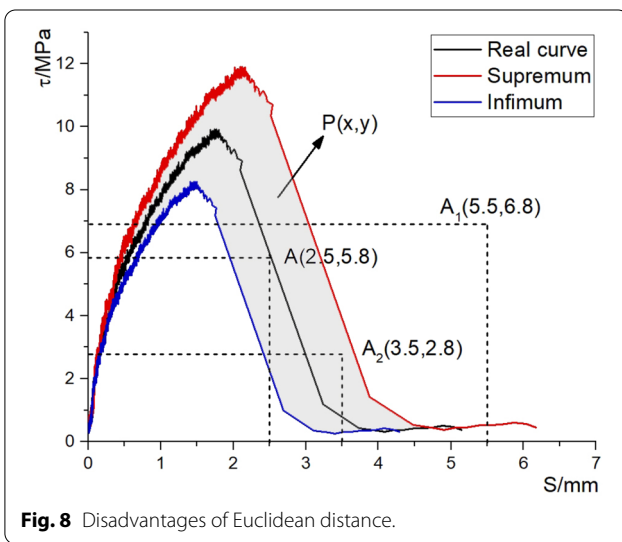
4.1 Training and Test Results of the MMN

Since the number of samples is too large to display them all, the training results of the MMN in the target tasks are randomly selected, as shown in Fig. 10. R^2 (coefficient of determination) was introduced in this paper to evaluate the prediction results, which can be calculated as follows:

$$R^2 = \begin{cases} 0 & R^2 \leq 0 \\ 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} & 0 \leq R^2 \leq 1 \\ 1 & R^2 \geq 1 \end{cases}$$

y_i is the i th test value, \hat{y}_i is the i th predicting value, and \bar{y} is the average value of the test values.

Fig. 10 shows that for the three tasks, the R^2 of the training set ranges from 0.8973 to 0.9913, and the R^2 of the test set ranges from 0.9181 to 0.9902. Therefore, both the training and test sets of the MMN perform well, and only some outliers appear in the descending stages of some specimens. The explanation for this phenomenon



is that hypothesis (4) in Sect. 2.1 may introduce some errors.

Therefore, the MMN prediction method can not only be used to solve the SRRC task but can also be used to solve the other bond–slip prediction tasks, which means that the MMN method has a high generalization level. In the next section, the MMN method will be further compared with the existing prediction methods to verify its reliability.

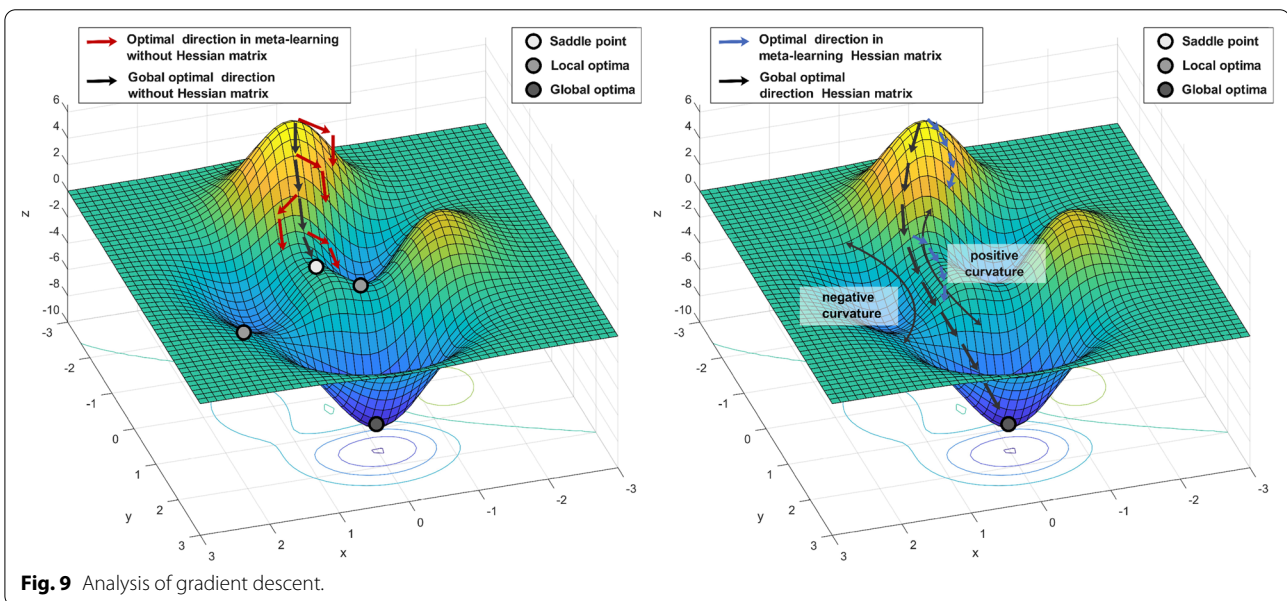
4.2 Algorithm Comparison

In Fig. 11, the results of eight methods were compared with the test results. Both the R^2 and mean square error (MSE) are introduced in this section to evaluate the prediction results. The MSE can be calculated as follows:

$$MSE = \frac{\sum (y_i - \hat{y}_i)^2}{m}$$

y_i is the i th test value, \hat{y}_i is the i th predicting value, and “ m ” is the sample number. If R^2 is closer to 1 and MSE is closer to 0, the final result indicates a better result. In addition, the success rate can be determined as the ratio of the number of samples with $R^2 > 0$ and the total sample number.

Because the fitting objects of the samples are ordinary rebar and ordinary concrete, Model (6) (CAoB 2010) and Model (8) (Wu & Zhao, 2013) have low prediction accuracy for the three special tasks: SRRC task, task-A (coated-rebar-task) and task-B (geopolymer concrete task). In addition, training results from other model are



compared in Table 5. Table 5 shows that the MMN performs the best of all the compared models in almost all cases. The prediction accuracy of the four-stage model (83.33% success rate) and MAML (58.33% success rate) is only second to that of the MMN, which has a certain anti-interference ability. However, when there are outlier samples in the prediction task (e.g., #320 and #66), the accuracy of the test set decreases. The other models perform poorly and are almost unsuccessful in some tasks (success rate less than 50%).

Taking Sample #320 as an example, the causes of the above situation were further explained: #320 sample

(Mo & Chan, 1996) belongs to the coated-rebar-task, but it comes from the other batch of concrete (Wang et al., 2020) with a curve that is quite different from those of other samples in this task. Due to the existence of the #320 sample, the models of overtraining small datasets fail to predict the curve of this sample. Moreover, if there are other similar samples in the task as the training set, the process of updating the parameters in the training set will sacrifice the overall accuracy to take these outlier samples into account. Therefore, only the MMN model and MAML model, which are based on task learning, have certain anti-interference effects. However, when

Table 4 Description of calculation models.

References	Model	Data set	Model details
This paper	MMN (Model 1)	Bondslipnet + SRRC-Net or data set from task A and B	(1) Meta-learning with Bondslipnet (2) Fine-tune with data from target task (3) Multi-tasking learning framework (4) Mahalanobis distance loss function (5) FRN normalization + dropout + L2 (6) Consider the Heissel matrix
Finn et al. (2017)	MAML (Model 2)	Bondslipnet + SRRC-Net or data set from task A and B	(1) Meta-learning with Bondslipnet (2) Fine-tune with data from target task (3) Using traditional DNN framework
Amaratunga, (2021)	Fine-tune (Model 3)	Bondslipnet + SRRC-Net or data set from task A and B	(1) Pretrain with Bondslipnet samples (2) Fine-tune with data from target task (3) Traditional DNN
Naderpour et al. (2019), Zhou et al. (2020a, 2020b)	DNN (Model 4)	Bondslipnet + SRRC-Net or data set from task A and B	DNN framework with Bondslipnet and data from target task
Naderpour et al. (2019), Zhou et al. (2020a), (2020)	DNN (Model 5)	SRRC-Net or data set from task A and B	DNN framework with data from target task
CAoB 2010	Four-linear-stages model (Model 6)	Data from this reference	$\tau = \begin{cases} (100f_t/d)s & 0 \leq s < 0.025d \\ 100f_t/3d(s - 0.025d) + 2.5f & 0.025d \leq s < 0.04d \\ -2f_t/0.51d(s - 0.04d) + 3f & 0.04d \leq s < 0.55d \\ f_t & s \geq 0.55d \end{cases}$
CAoB 2010	Four-linear-stages model (Model 7)	SRRC-Net or data set from task A and B	$\tau = \begin{cases} (b_1f_t/a_1d)s & 0 \leq s < a_1d \\ (b_2 - b_1)f_t/(a_2 - a_1)d(s - a_1d) + b_1f & a_1d \leq s < a_2d \\ (b_3 - b_2)f_t/(a_3 - a_2)d(s - a_2d) + b_2f & a_2d \leq s < a_3d \\ b_3f_t & s \geq a_3d \end{cases}$
Wu and Zhao (2013)	Three-nonlinear-stages model (Model 8)	Data from this reference	$\tau = \frac{\tau_{\max}(e^{Bs} - e^{Ds})}{[e^{-B \ln(B/D)} - e^{-D \ln(B/D)}]}$ $K_{co} = \frac{c}{d}, K_{st} = \frac{A_{st}}{nS_{st}d}$ $K_{sj} = \left(\frac{d}{25.4}\right)^{0.2-1.8\frac{H_{rib}}{S_{rib}}} \text{ or } K_{sj} = d^{-0.4}$ $K = (K_{co} + 33K_{st})K_{sj}$ $\frac{\tau_{\max}}{\sqrt{f_c}} = \frac{2.5}{1 + 3.1e^{-0.47(K_{co} + 33K_{st})}}$ $B = \frac{0.0254 + K_{st}}{-0.0232 - 8.34K_{st}}$ $D = 3 \ln \left(\frac{0.7315 + K}{5.176 + 0.333K} - 0.13 \right) - 3.375$

The difference between Model 6 and Model 7 is that Model 6 directly adopts the formulas (CAoB 2010), and Model 7 re-regresses the data from the target task.

the number or difference of such samples increases, the MAML model using the Euclidean distance to measure error also has low accuracy or even fails because it does not fully learn the relationship between the sample dimensions (e.g., the task $A_{\text{squ}}B_{\text{pla}}C_{\text{nwat}}D_{\text{sti}}E_{\text{nor}}$ has outliers: #66, #75, and #78 in the training set). The explanation for the low prediction accuracy of MAML compared to MMN can be further discussed as follows. On the one hand, the loss function of the MMN is the Mahalanobis distance modified by the single-layer perceptron replacing the Euclidean distance. Therefore, the MMN model further learns the distribution of the output data in two dimensions. On the other hand, as described in Sect. 3.3.2, the Hessian matrix correction term was used

in the MMN model to help search the parameter space of the mapping mode, which will help the model jump out of the local minimum and approach as close to the global minimum as possible.

In summary, the MMN can be considered to have better applicability and accuracy for predicting the bond–slip performance. In the next section, the reason why the MMN model has advantages will be further verified.

4.3 Advantages of the MMN

4.3.1 Smaller Loss

The training loss of the five neural networks is shown in Fig. 12. The figure shows that the loss of the MMN

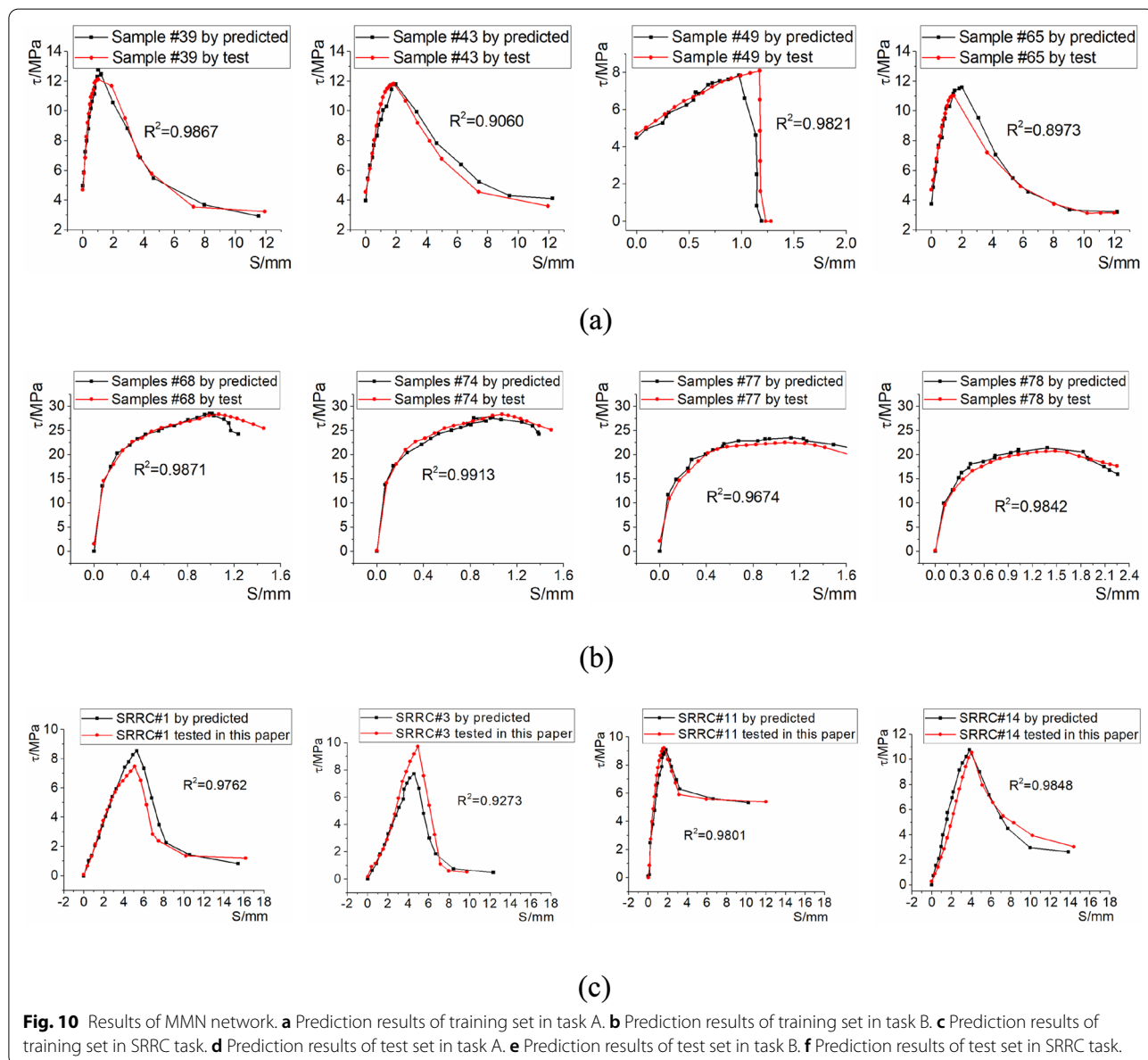


Fig. 10 Results of MMN network. **a** Prediction results of training set in task A. **b** Prediction results of training set in task B. **c** Prediction results of training set in SRRC task. **d** Prediction results of test set in task A. **e** Prediction results of test set in task B. **f** Prediction results of test set in SRRC task.

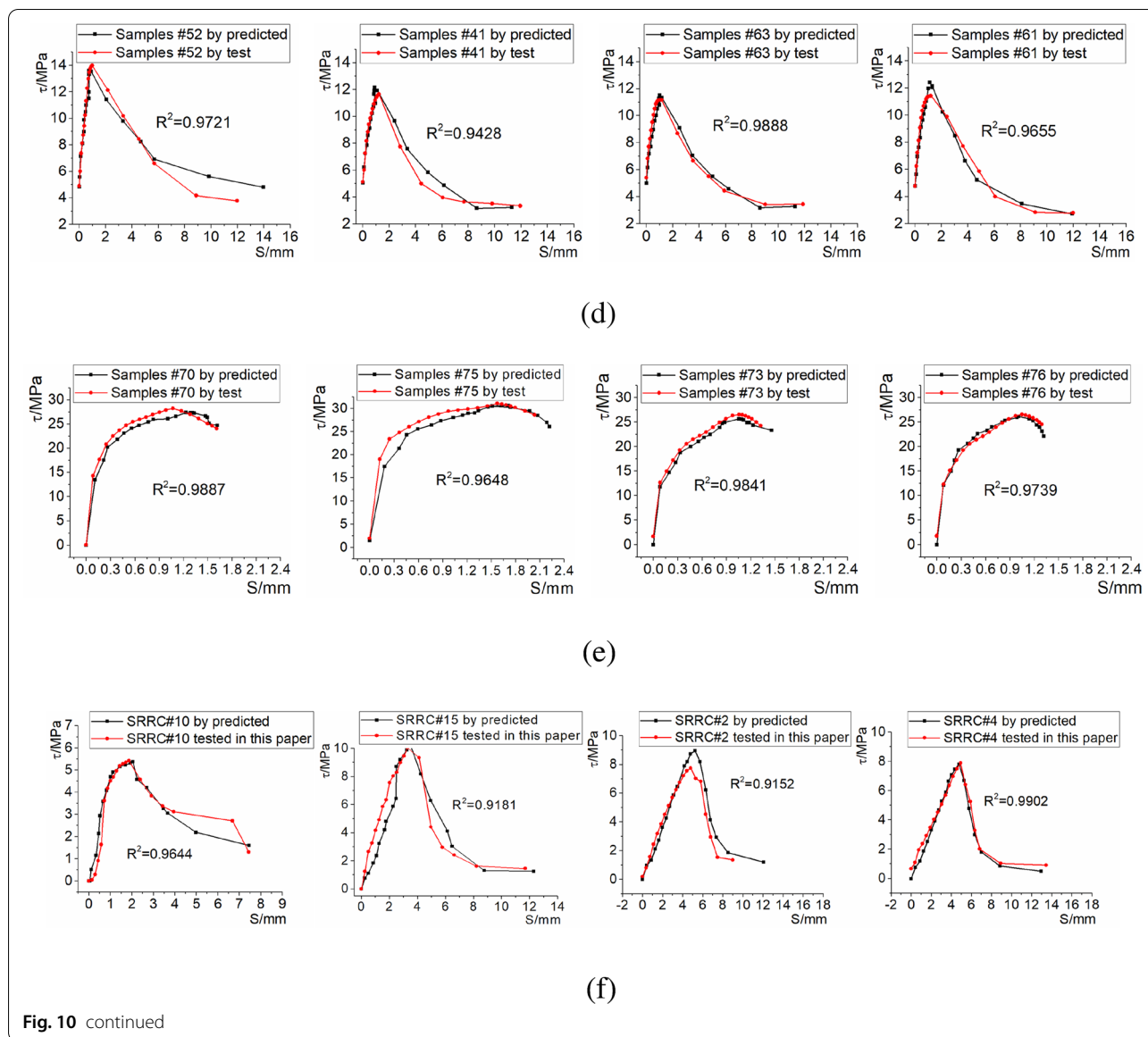


Fig. 10 continued

is smaller than those of the other models. The loss of the MMN converges at approximately 20 epochs and gradually tends to be flat after 40 epochs.

However, other networks still fluctuate obviously until the later stage. The fine-tune model even jumps out of the convergence range in the later stage of training, which is due to the sample shuffling during each iteration. For the fine-tune model, its anti-interference is not outstanding, and the distribution of data between each task is quite different, which causes the loss to be nonconvergent.

4.3.2 Better Initial Value

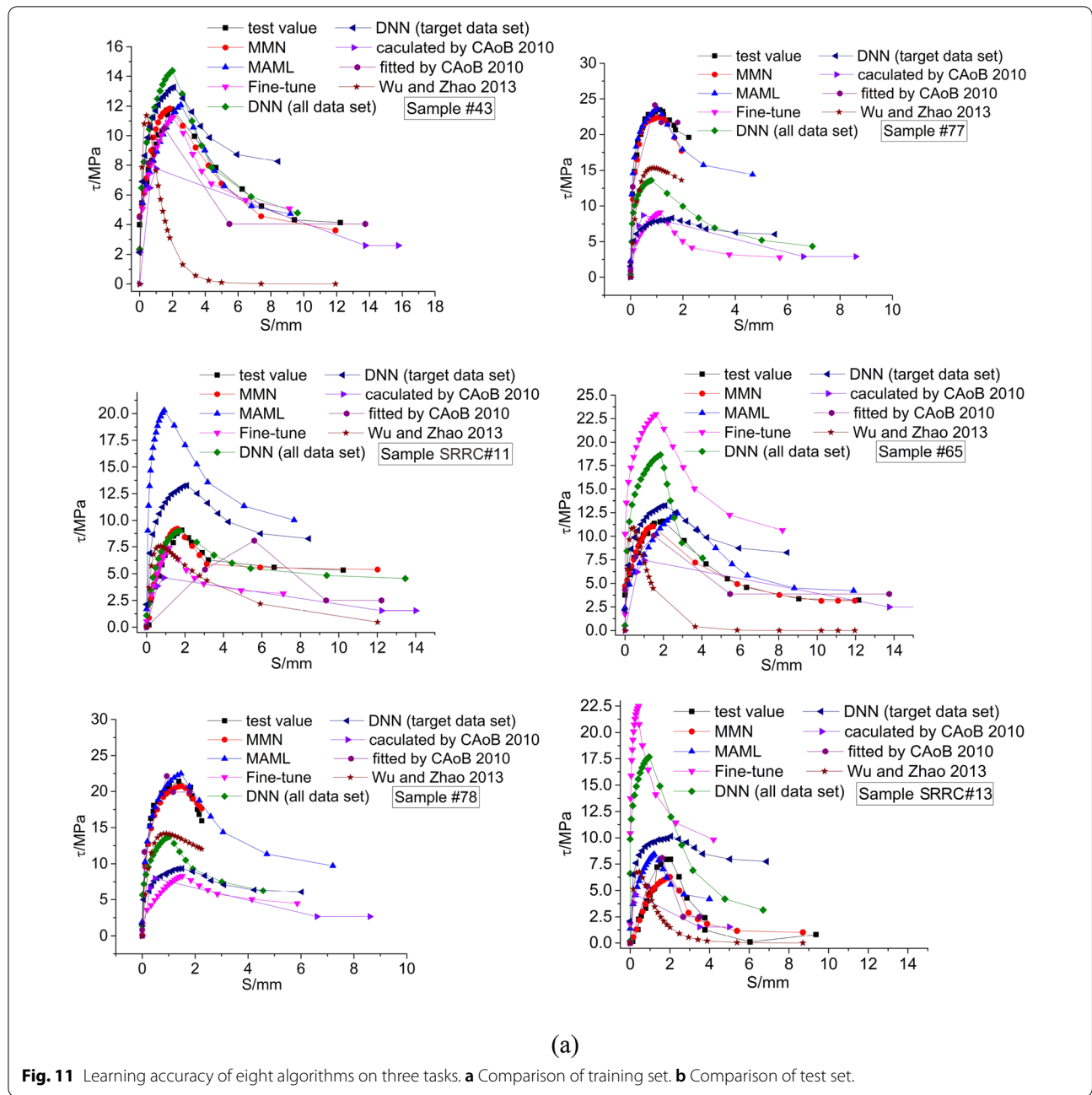
The meta-learning process of two randomly selected samples, Sample #185 (from task $A_{\text{squ}}B_{\text{pla}}C_{\text{arep}}D_{\text{nsti}}E_{\text{cor}}$)

and Sample #297 (from task $A_{\text{squ}}B_{\text{pla}}C_{\text{arep}}D_{\text{nsti}}E_{\text{nor}}$), in the MMN is shown in Fig. 13. After 50 epochs, even though the training results of meta-learning are directly used for prediction, a certain accuracy can be achieved. It should be noted that the DNN prediction models train the data based on the errors between the points, which means that during the training process, there may be some points that do not satisfy the physical meanings (as the bond strength increases, slip decreases at a certain level). As more epochs were carried out, this phenomenon was gradually alleviated. However, some limitations based on the model itself will make the above phenomenon unable to completely disappear. As shown in Fig. 13, the results of Sample #185 show that the MMN model makes a good

contribution to solving this problem, and those of Sample #297 show that the MMN model helps alleviate this phenomenon. The explanation for the result of Sample #297 can be explained as follows. It should be noted that Fig. 13 was used to present the training process in the meta-learning period, which is not the final result. In this process, the task with sample #297 has not been trained using Fine-tune, which means the data distribution of the samples in this task has not been trained specifically.

That will not effectively play the role of the modified Mahalanobis distance in the MMN model.

The prediction results of the MMN, MAML, fine-tuning and DNN (all datasets) algorithms after the initial parameter optimization process are compared, as shown in Fig. 14. The training results of the MMN before fine-tuning are even better than those of the DNN that has completed the training in the overall trend.



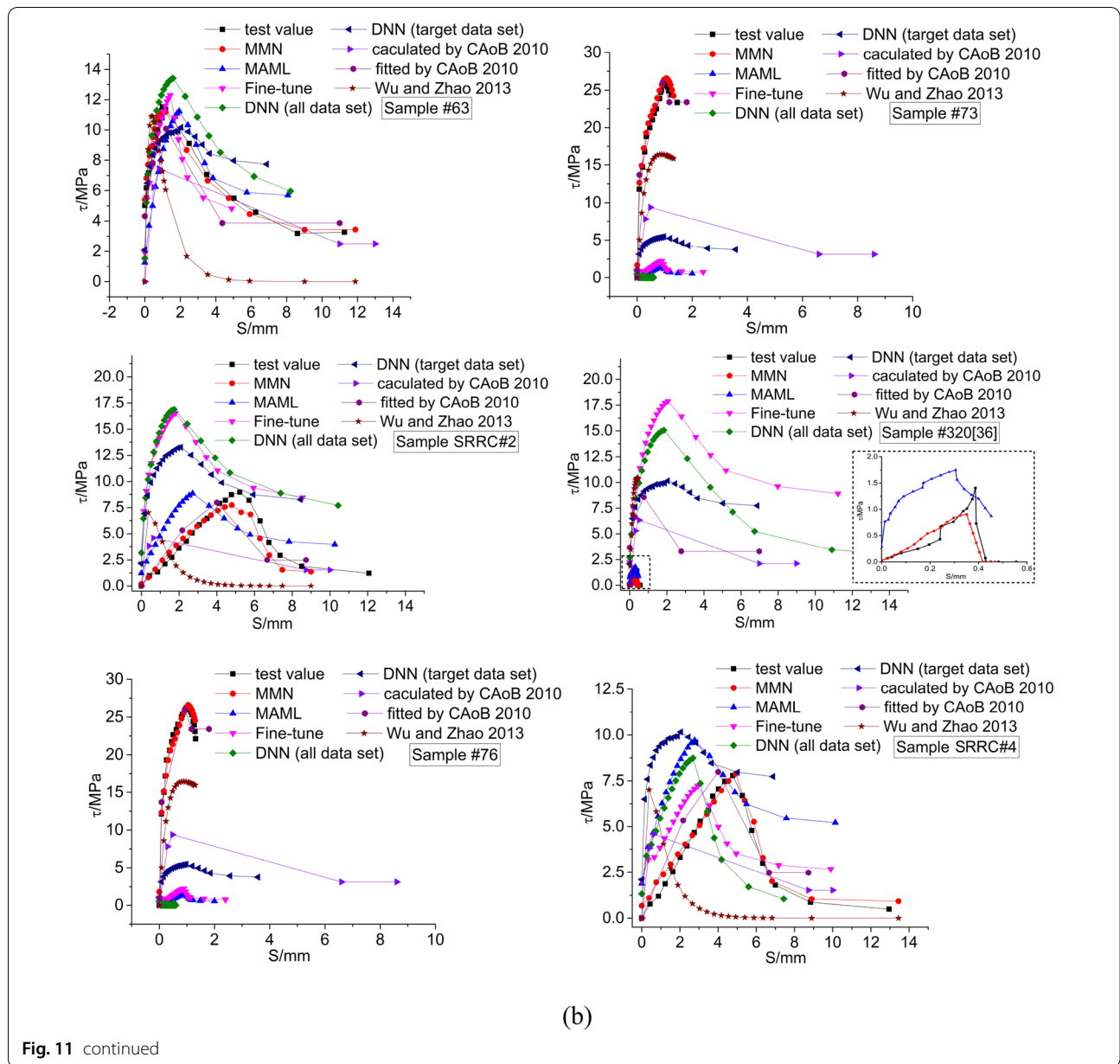


Fig. 11 continued

4.3.3 Easier Gradient Descent

The relationship between the parameters (randomly selected two-dimensional) of the five models and the global loss is shown in Fig. 15, and the gradient search direction is marked in the figure. The surface of the MMN method is relatively smooth because the distribution of the sample itself was learned when considering the loss function. Moreover, the MMN takes the Hessian matrix into consideration so that with the curvature variation characteristics of the loss function, the search direction of the optimal path is clearer. The global loss of other models has many local minimum points or saddle

points that are difficult to jump out. Even if the model introduces noise, in most cases, it may jump out of the original extreme point and may even fall into a worse extreme point. Therefore, the MMN can be considered easier for gradient descent.

4.4 Parameter Analysis About Section Size of Rebar

In this section, based on the database and model established in this paper, the change in the ultimate bond strength of SRRC and C30 ordinary reinforced concrete with the rebar section size (under the same other design

Table 5 Details of the results of each model.

Tasks	No.	Label	Prediction accuracy ranking (maximum representative optimal)
Asqu Bpla Cnor Dnsti Ecoa	#43	Train	Successful models: MAML(MSE=0.49, $R^2=0.93$) > Fine-tune (MSE=0.61, $R^2=0.92$) > MMN (MSE=0.71, $R^2=0.91$) > Four-stage model (MSE=0.76, $R^2=0.90$) > DNN (all data set (MSE=2.21, $R^2=0.71$)) > DNN (target data set (MSE=3.31, $R^2=0.56$)) Failed models:none
	#65	Train	Successful models: four-stage model (MSE=0.40, $R^2=0.97$) > MMN (MSE=1.19, $R^2=0.90$) > MAML (MSE=2.48, $R^2=0.79$) > DNN(target data set (MSE=10.391, $R^2=0.10$)) Failed models: DNN (all data set (MSE=32.73, $R^2=0$)), Fine-tune(MSE=59.82, $R^2=0$)
	#63	Test	Successful models: MMN(MSE=0.10, $R^2=0.99$) > Four-stage model(MSE=1.28, $R^2=0.85$) > MAML(MSE=2.87, $R^2=0.66$) > DNN(target data set (MSE=3.79, $R^2=0.56$)) > DNN (all data set(MSE=3.82, $R^2=0.55$)) > Fine-tune (MSE=3.87, $R^2=0.55$) > Failed models: none
	#320	Test	Successful models: MMN (MSE=0.02, $R^2=0.59$) > MAML (MSE=0.33, $R^2=0.10$) Failed models: four-stage model(MSE=15.94, $R^2=0$), DNN (target data set (MSE=37.84, $R^2=0$)), DNN (all data set (MSE=62.56, $R^2=0$)), Fine-tune (MSE=94.18, $R^2=0$)
Asqu Bpla Cnwat Dsti Enor	#77	Train	Successful models: MMN(MSE=0.40, $R^2=0.97$) > MAML (MSE=1.49, $R^2=0.88$) > Four-stage model (MSE=1.66, $R^2=0.87$) > DNN (all data set (MSE=51.75, $R^2=0$)) Failed models: DNN (target data set(MSE=81.85, $R^2=0$)), Fine-tune (MSE=91.44, $R^2=0$)
	#78	Train	Successful models: MMN(MSE=0.19, $R^2=0.98$) > Four-stage model (MSE=0.76, $R^2=0.94$) > MAML(MSE=4.42, $R^2=0.62$) Failed models:DNN (all data set (MSE=28.17, $R^2=0$)), DNN (target data set (MSE=51.84, $R^2=0$)), Fine-tune (MSE=67.40, $R^2=0$)
	#73	Test	Successful models:MMN (MSE=0.29, $R^2=0.98$) > Four-stage mode l(MSE=1.27, $R^2=0.93$) Failed models: DNN (target data set (MSE=161.68, $R^2=0$)), Fine-tune (MSE=223.61, $R^2=0$), MAML (MSE=234.6, $R^2=0$), DNN (all data set (MSE=252.98, $R^2=0$))
	#76	Test	Successful models:MMN (MSE=0.47, $R^2=0.97$) > Four-stage model (MSE=1.22, $R^2=0.93$) Failed models: DNN (target data set (MSE=160.70, $R^2=0$)), Fine-tune (MSE=222.5, $R^2=0$), MAML (MSE=233.4, $R^2=0$), DNN (all data set (MSE=251.73, $R^2=0$))
SRRC task	SRRC#11	Train	Successful models: MMN (MSE=0.14, $R^2=0.98$) > DNN (all data set (MSE=0.80, $R^2=0.88$)) > Fine-tune (MSE=1.94, $R^2=0.72$) Failed models: four-stage model (MSE=10.29, $R^2=0$), DNN (target data set (MSE=10.77, $R^2=0$)), MAML (MSE=43.45, $R^2=0$)
	SRRC#13	Train	Successful models:MMN(MSE=0.44, $R^2=0.89$) > Four-stage model (MSE=0.60, $R^2=0.86$) Failed models: MAML (MSE=5.05, $R^2=0$), DNN (target data set (MSE=14.36, $R^2=0$)), DNN (all data set (MSE=41.61, $R^2=0$)), Fine-tune (MSE=98.12, $R^2=0$)
	SRRC#2	Test	Successful models:MMN (MSE=0.49, $R^2=0.92$) > Four-stage model (MSE=0.50, $R^2=0.91$) > MAML (MSE=1.87, $R^2=0.68$) Failed models: DNN (target data set (MSE=21.94, $R^2=0$)), Fine-tune (MSE=34.67, $R^2=0$), DNN (all data set (MSE=35.37, $R^2=0$))
	SRRC#4	Test	Successful models: MMN (MSE=0.07, $R^2=0.99$) > Four-stage model (MSE=0.62, $R^2=0.92$) > Fine-tune (MSE=1.84, $R^2=0.76$) > DNN (all data set (MSE=3.94, $R^2=0.48$)) > MAML (MSE=5.87, $R^2=0.22$) Failed models; DNN (target data set (MSE=15.59, $R^2=0$))

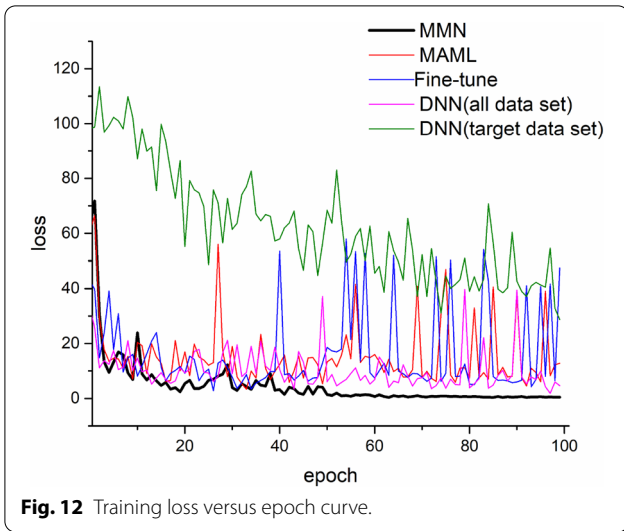
The failed models are those where the R^2 values are approximately equal to 0.

conditions: concrete cover thickness $c=5$ d, anchorage length $L=5$ d) is analyzed, as shown in Fig. 16a. In addition, the variation in the relative slip “ S_u ” corresponding to the ultimate bond strength with the section size of the rebar is also analyzed in Fig. 16b. The ultimate bond strength of SRRC and C30 ordinary reinforced concrete with rebar section size changes according to the same general rule, which has peak values of $d=12-14$ mm. Compared with C30 ordinary reinforced concrete, the ultimate bond strength of SRRC decreased on average by approximately 33.59% (coefficient of variation is about 0.28, standard deviation is 9.45%). For S_{u^*} , the slip of C30

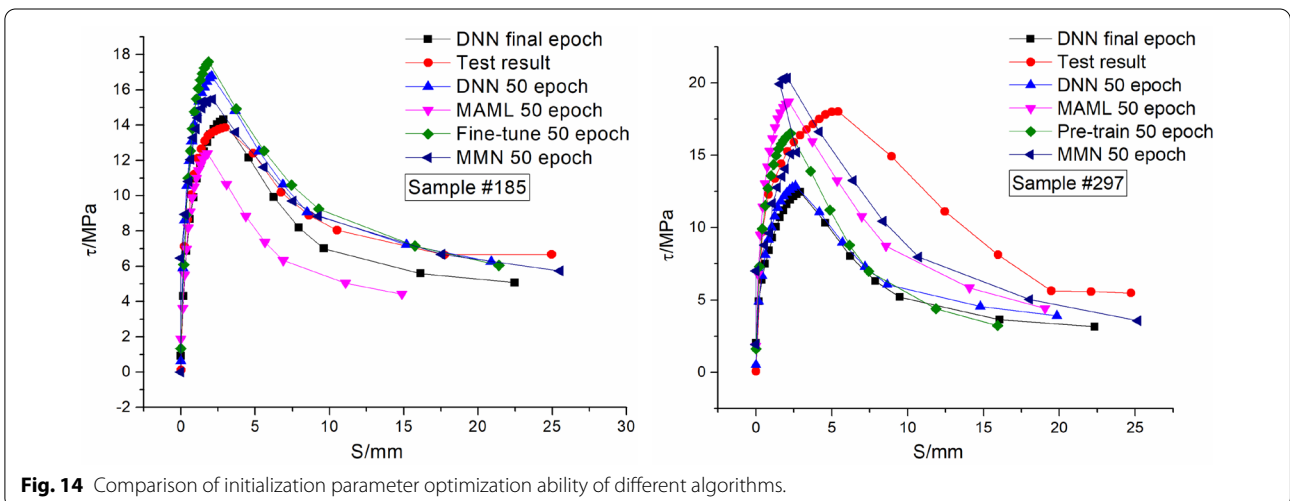
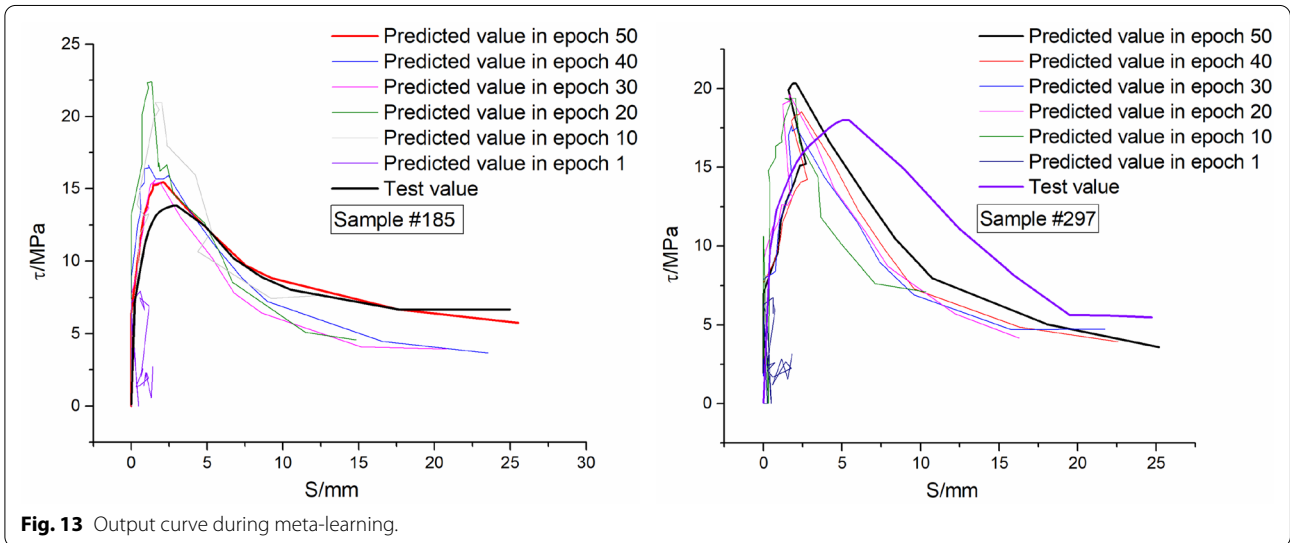
ordinary reinforced concrete is far less than that of SRRC, and the average decrease is about 74.63% (coefficient of variation is about 0.15, standard deviation is 11.35%).

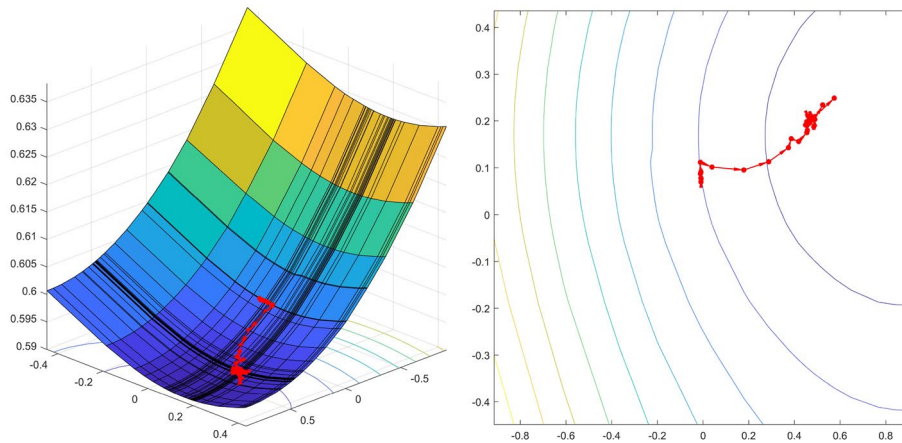
5 Conclusion

Based on more than 2000 tests performed by 36 references, a large database BondSlipNet containing 558 samples was built, and 16 groups of bond–slip tests on SRRC reinforced concrete were performed, based on which the SRRC-Net database was established. These two databases can be directly used for researcher to study the data-driven method for the bond–slip

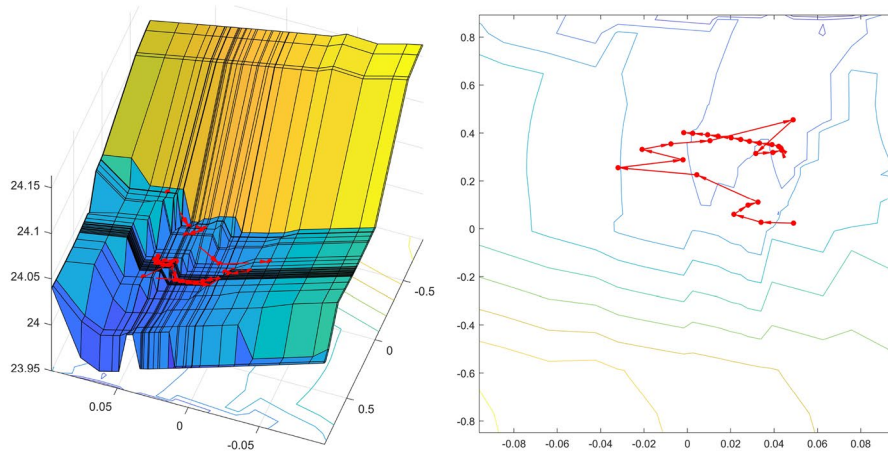


problem. Then, a neural network MMN algorithm was proposed, which is suitable for the prediction task with a small sample size and implicit relationship between bond–slip characteristics and curves. This algorithm had sufficient accuracy for the SRRC task and the other two randomly selected tasks and was compared with the existing seven algorithms to verify its accuracy, convergence ability and generalization level. The well-trained MMN neural network can be used directly in real engineering for each task involved in BondSlip-Net database. The applicable design parameter range in the MMN model is outlined as shown in the second column of Table 2, which covers most of the common parameter conditions of reinforced concrete bond–slip behavior. Furthermore, if the prediction tasks are out of the scope of the BondSlipNet database (the parameter

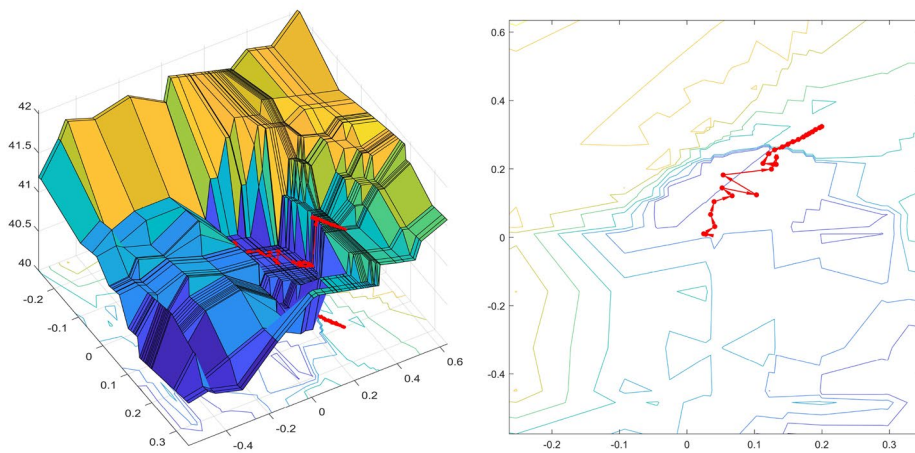




(a) Loss function of the MMN algorithm

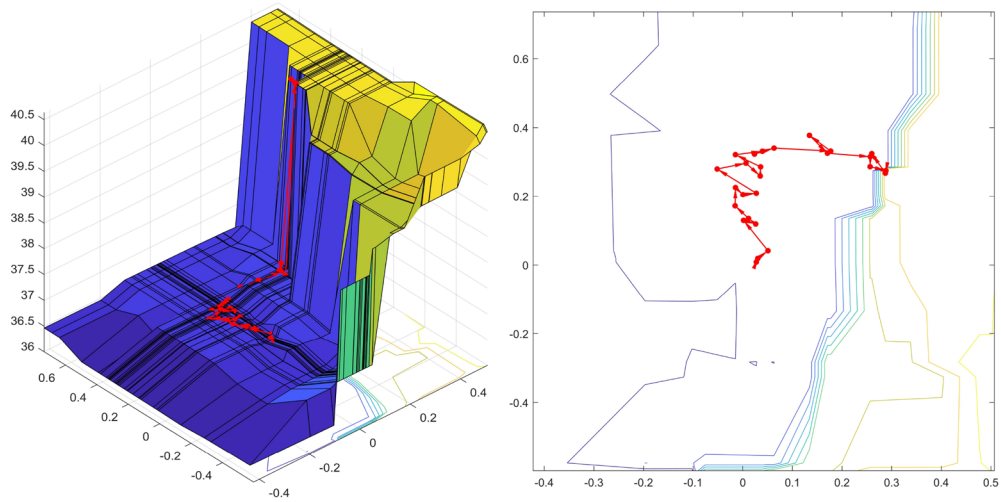


(b) Loss function of the MAML algorithm

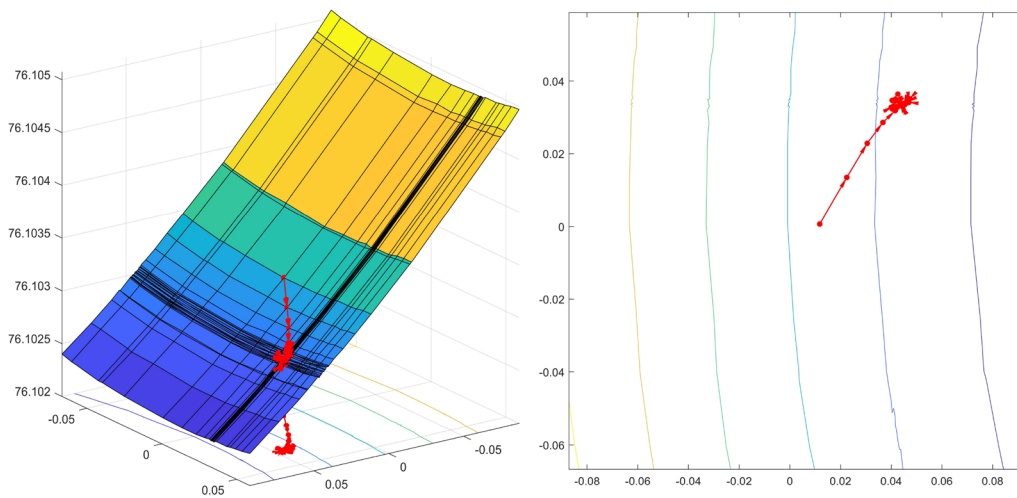


(c) Loss function of the Fine-tune algorithm

Fig. 15 Gradient descent of different algorithms. **a**) Loss function of the MMN algorithm. **b**) Loss function of the MAML algorithm. **c**) Loss function of the Fine-tune algorithm. **d**) Loss function of the DNN (all data set) algorithm. **e**) Loss function of the DNN (target data set) algorithm.



(d) Loss function of the DNN (all data set) algorithm



(e) Loss function of the DNN (target data set) algorithm

Fig. 15 continued

range exceeds the values listed in the second column of Table 2), the MMN algorithm and the built databases can also be used to transfer the prior knowledge for the new tasks. The following conclusions and prospects can be drawn:

(1) The concrete proportion and section shape of the rebar in SRRC reinforced concrete are significantly different from those of modern reinforced concrete, leading to different bond-slip performances. Overall, the bond strength between SRRC rebar and concrete is relatively

low (compared with C30 ordinary reinforced concrete, the average strength decline is approximately 33.59% with a coefficient of variation of approximately 0.28 and a standard deviation of 9.45%), and it is difficult to explicitly develop a unified bond-slip model.

(2) The existing prediction models for bond-slip often only aim at a specific prediction task, which are difficult to apply directly when transferring to new tasks. The traditional DNN model is not sensitive to new tasks because of overfitting the characteristics of the sample. The

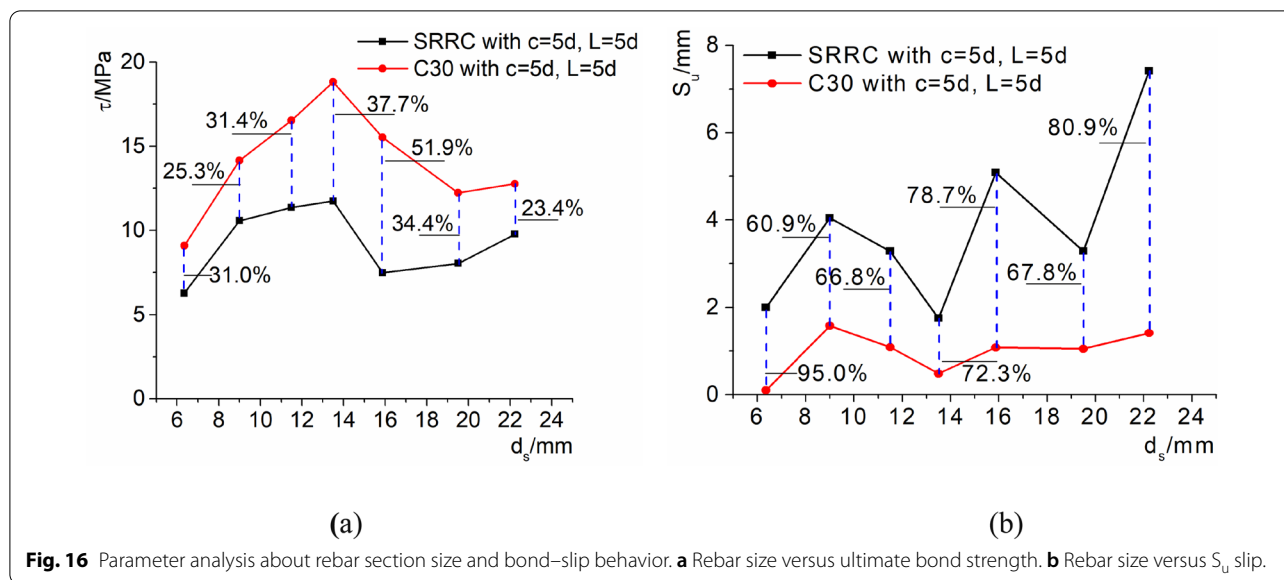


Fig. 16 Parameter analysis about rebar section size and bond-slip behavior. **a** Rebar size versus ultimate bond strength. **b** Rebar size versus S_u slip.

four-linear-stage model and three-nonlinear-stage model require many tests on new tasks to re-regress the parameters and even need to reselect the fitting function, which greatly increases the test cost and research effort.

(3) The MMN algorithm proposed in this paper possesses the characteristics of MAML algorithm learning based on tasks and uses the learned single-layer perceptron-modified Mahalanobis distance loss to replace the MSE loss. At the same time, the network is rewritten as a multitask learning framework, and the role of the Hessian matrix is considered. The calculation results and mathematical analysis show that these improvements contribute to the prediction of the bond-slip model.

The accuracy of the MMN exceeds that of some existing algorithms, but it also falls into a certain bottleneck. For some specific samples, there will be some outlier points in the descent stage. In addition, although the overall trend of the model is close to that of the test, the distance between the predicted points and label points can be improved. This can be achieved by further improving the labels, expanding the database capacity and adjusting the local network algorithm.

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s40069-022-00530-y>.

Additional file 1. A large pretraining database for bond-slip model.

Additional file 2. A small fine-tune database for bond-slip model.

Acknowledgements

The authors gratefully acknowledge sponsorship by the Jiangsu Provincial Key Research and Development Program (Grant No. BE2017717) and Scientific Research Foundation of Graduate School of Southeast University (YBPY2101).

Author contributions

CZ contributed to the conceptualization, methodology, software, formal analysis, and writing—original draft. QC contributed to validation, supervision, project administration, funding acquisition and writing—review and editing. AS contributed to resources, data curation, writing—review and editing and visualization. YL contributed to resources, investigation and writing—review and editing. HW contributed to resources and investigation. All authors read and approved the final manuscript.

Authors' information

Chengwen Zhang: Ph.D., School of Architecture, Southeast University, Nanjing, Jiangsu, 210096, China, ORCID: <https://orcid.org/0000-0002-5938-7559>. E-mail: zhang1chengwen@163.com.

Qing Chun: Professor, School of Architecture, Southeast University, Nanjing, Jiangsu, 210096, China (corresponding author). E-mail: cqnj1979@163.com.

Ao Sun: Ph.D. Candidate, School of Civil Engineering, Southeast University, Nanjing, Jiangsu, 210096, China, E-mail: 821,031,305@qq.com.

Yijie Lin: Ph.D., School of Architecture, Southeast University, Nanjing, Jiangsu, 210096, China, E-mail: linyijie1223@163.com.

Haoyu Wang: Ph.D. Candidate, School of Architecture, Southeast University, Nanjing, Jiangsu, 210096, China, E-mail address: mathewwhy@163.com.

Funding

This research was support by the Jiangsu Provincial Key Research and Development Program (Grant No. BE2017717) and Scientific Research Foundation of Graduate School of Southeast University (YBPY2101).

Availability of data and materials

All data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Architecture, Southeast University, Nanjing 210096, China. ²School of Civil Engineering, Southeast University, Nanjing 210096, China.

Received: 10 December 2021 Accepted: 16 April 2022

Published online: 29 July 2022

References

- Amaratunga, T. (2021). Transfer learning. *Deep Learning on Windows* https://doi.org/10.1007/978-1-4842-6431-7_7
- Béton, C. C. E.-I. d. (1990). CEB-FIP MODEL CODE. 1990. London. <https://doi.org/10.1680/ceb-fipmc1990.35430>
- Biscaia, H. C., & Soares, S. (2020). Adherence prediction between ribbed steel rebars and concrete: A new perspective and comparison with codes. *Structures*, 25, 979–999. <https://doi.org/10.1016/j.istruc.2020.04.019>
- Cai, B., Xu, L. F., & Fu, F. (2019). Shear resistance prediction of post-fire reinforced concrete beams using artificial neural network. *Int J Concr Struct Mater*, 13, 46. <https://doi.org/10.1186/s40069-019-0358-8>
- Cai, J. M., Pan, J. L., Tan, J. W., & Li, X. P. (2020). Bond behaviours of deformed steel rebars in engineered cementitious composites (ECC) and concrete. *Construction and Building Materials*, 252, 12. <https://doi.org/10.1016/j.conbuildmat.2020.119082>
- Coccia, S., Imperatore, S., & Rinaldi, Z. (2016). Influence of corrosion on the bond strength of steel rebars in concrete. *Materials and Structures*, 49(1–2), 537–551. <https://doi.org/10.1617/s11527-014-0518-x>
- Finn, C., Abbeel, P., and Levine, S. (2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks." *arXiv e-prints*, arXiv:1703.03400.
- Fu, C., Fang, D., Ye, H., Huang, L., & Wang, J. (2021). Bond degradation of non-uniformly corroded steel rebars in concrete. *Engineering Structures*. <https://doi.org/10.1016/j.engstruct.2020.111392>
- Gao, X., Li, N., & Ren, X. (2019). Analytic solution for the bond stress-slip relationship between rebar and concrete. *Construction and Building Materials*, 197, 385–397. <https://doi.org/10.1016/j.conbuildmat.2018.11.206>
- Hou, L. J., Guo, S., Zhou, B. X., Xu, S. L., & Chen, D. (2018). Bond-slip behavior of corroded rebar embedded in ultrahigh toughness cementitious composite. *Journal of Materials in Civil Engineering*, 30(7), 12. [https://doi.org/10.1061/\(asce\)mt.1943-5533.0002324](https://doi.org/10.1061/(asce)mt.1943-5533.0002324)
- Islam, K., Billah, A. H. M. M., Chowdhury, M. M. I., & Ahmed, K. S. (2020). Exploratory study on bond behavior of plain and sand coated stainless steel rebars in concrete. *Structures*, 27, 2365–2378. <https://doi.org/10.1016/j.istruc.2020.07.039>
- Jang, Y., Ahn, Y., & Kim, H. Y. (2019). Estimating compressive strength of concrete using deep convolutional neural networks with digital microscope images. *J Comput Civil Eng*. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000837](https://doi.org/10.1061/(asce)cp.1943-5487.0000837)
- Kaffetzakis, M. I., & Papanicolaou, C. G. (2016). Bond behavior of reinforcement in lightweight aggregate self-compacting concrete. *Construction and Building Materials*, 113, 641–652. <https://doi.org/10.1016/j.conbuildmat.2016.03.081>
- Kakavand, M. R. A., Sezen, H., & Taciroglu, E. (2021). Data-driven models for predicting the shear strength of rectangular and circular reinforced concrete columns. *Journal of Structural Engineering*, 147(1), 12. [https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0002875](https://doi.org/10.1061/(ASCE)ST.1943-541X.0002875)
- Khaksefidi, S., Ghalehnovi, M., & de Brito, J. (2021). Bond behaviour of high-strength steel rebars in normal (NSC) and ultra-high performance concrete (UHPC). *Journal of Building Engineering*. <https://doi.org/10.1016/j.jobe.2020.101592>
- Koulouris, K., & Apostolopoulos, C. (2020). An experimental study on effects of corrosion and stirrups spacing on bond behavior of reinforced concrete. *Metals*, 10(10), 14. <https://doi.org/10.3390/met10101327>
- Leibovich, O., Yankelevsky, D. Z., & Dancygier, A. N. (2020). The effect of local rebar geometry on global strain measurements in pull out tests. *Structural Concrete*, 21(1), 413–427. <https://doi.org/10.1002/suco.201800358>
- Li, C., Miao, L. C., You, Q., Lei, L. J., Liang, X. D., & Wang, J. J. (2020). Experimental investigation on the bond behavior of deformed bars in a class of structural expanded polystyrene (EPS) concrete. *KSCCE Journal of Civil Engineering*, 24(12), 3797–3809. <https://doi.org/10.1007/s12205-020-0167-2>
- Li, X. Y., Zhang, J. W., Liu, J., & Cao, W. L. (2019). Bond behavior of spiral ribbed ultra-high strength steel rebar embedded in plain and steel fiber reinforced high-strength concrete. *KSCCE Journal of Civil Engineering*, 23(10), 4417–4430. <https://doi.org/10.1007/s12205-019-2449-0>
- Liu, X., Liu, Y., Wu, T., & Wei, H. (2020). Bond-slip properties between lightweight aggregate concrete and rebar. *Construction and Building Materials*, 255, 13. <https://doi.org/10.1016/j.conbuildmat.2020.119355>
- Metelli, G., & Pizzari, G. A. (2014). Influence of the relative rib area on bond behaviour. *Magazine of Concrete Research*, 66(6), 277–294. <https://doi.org/10.1680/mac.13.00198>
- Mo, Y. L., & Chan, J. (1996). Bond and slip of plain rebars in concrete. *Journal of Materials in Civil Engineering*, 8(4), 208–211. [https://doi.org/10.1061/\(ASCE\)0899-1561\(1996\)8:4\(208\)](https://doi.org/10.1061/(ASCE)0899-1561(1996)8:4(208))
- Nabilah, A. B., Koh, C. G., Izian, A. K., et al. (2020). Development of finite element analysis for intermediate length coupling beams considering bond-slip interface. *Int J Concr Struct Mater*, 14, 33. <https://doi.org/10.1186/s40069-020-00409-w>
- Naderpour, H., Mirrashid, M., & Nagai, K. (2019). An innovative approach for bond strength modeling in FRP strip-to-concrete joints using adaptive neuro-fuzzy inference system. *Engineering Computations*, 36(3), 1083–1100. <https://doi.org/10.1007/s00366-019-00751-y>
- Nguyen, H. A., Chang, T. P., Lee, P. H., & Shih, J. Y. (2020). Experimental investigation of bond-slip performance of reinforcement in two green concretes. *Journal of Materials in Civil Engineering*, 32(3), 7. [https://doi.org/10.1061/\(asce\)mt.1943-5533.0003029](https://doi.org/10.1061/(asce)mt.1943-5533.0003029)
- Paswan, R., Rahman, M. R., Singh, S. K., & Singh, B. (2020). Bond behavior of reinforcing steel bar and geopolymer concrete. *Journal of Materials in Civil Engineering*, 32(7), 12. [https://doi.org/10.1061/\(asce\)mt.1943-5533.0003237](https://doi.org/10.1061/(asce)mt.1943-5533.0003237)
- Pauletta, M., Rovere, N., Randl, N., & Russo, G. (2020). Bond-slip behavior between stainless steel rebars and concrete. *Materials*, 13(4), 17. <https://doi.org/10.3390/ma13040979>
- Prince, M. J. R., & Singh, B. (2013). Bond behaviour between recycled aggregate concrete and deformed steel bars. *Materials and Structures*, 47(3), 503–516. <https://doi.org/10.1617/s11527-013-0075-8>
- Prince, M. J. R., & Singh, B. (2015). Bond behaviour of normal- and high-strength recycled aggregate concrete. *Structural Concrete*, 16(1), 56–70. <https://doi.org/10.1002/suco.201300101>
- Qi, A., Liu, X., Xu, R., & Huang, Y. (2020). Bond behavior of steel reinforcement in concrete containing ferronickel slag and blast furnace slag powder. *Construction and Building Materials*. <https://doi.org/10.1016/j.conbuildmat.2020.120884>
- Rabi, M., Cashell, K. A., Shamass, R., & Desnerck, P. (2020). Bond behaviour of austenitic stainless steel reinforced concrete. *Engineering Structures*. <https://doi.org/10.1016/j.engstruct.2020.111027>
- Rafi, M. M. (2019). Study of bond properties of steel rebars with recycled aggregate concrete. experimental testing. *Strength of Materials*, 50(6), 937–950. <https://doi.org/10.1007/s11223-019-00042-3>
- Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F. A., Bengio, Y., and Courville, A. (2018). On the spectral bias of neural networks. *arXiv e-prints*, arXiv:1806.08734.
- Research C. A. o. B. (2010). *Code for design of concrete structures* (p. 439). China Architecture & Building Press.
- Rockson, C., Tamanna, K., Alam, M. S., & Rteil, A. (2020). Effect of cover on bond strength of structural concrete using commercially produced recycled coarse and fine aggregates. *Construction and Building Materials*, 255, 16. <https://doi.org/10.1016/j.conbuildmat.2020.119275>
- Singh, S., & Krishnan, S. Filter response normalization layer: eliminating batch dependence in the training of deep neural networks." Proc., 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11234–11243. <https://doi.org/10.1109/CVPR42600.2020.01125>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- Sun, T., Shao, Y., Li, X., Liu, P., & Huang, X. (2020). Learning sparse sharing architectures for multiple tasks. AAAI Conference on Artificial Intelligence.
- Vanneschi, L., Castelli, M., Scott, K., et al. (2018). Accurate high performance concrete prediction with an alignment-based genetic programming system. *Int J Concr Struct Mater*, 12, 72. <https://doi.org/10.1186/s40069-018-0300-5>
- Wang, H. L. (2016). Steel-concrete bond behaviour of self-compacting concrete with recycled aggregates. *Magazine of Concrete Research*, 68(13), 678–691. <https://doi.org/10.1680/jmacr.15.00143>
- Wang, X., Liu, Y., & Xin, H. (2021). Bond strength prediction of concrete-encased steel structures using hybrid machine learning method. *Structures*, 32, 2279–2292. <https://doi.org/10.1016/j.istruc.2021.04.018>
- Wang, X., Liu, Y., Yang, F., Lu, Y., & Li, X. (2019). Effect of concrete cover on the bond-slip behavior between steel section and concrete in SRC structures. *Construction and Building Materials*, 229, 116855. <https://doi.org/10.1016/j.conbuildmat.2019.116855>

- Wang, F. C., Wu, X. M., Guo, C., & Song, W. (2018). Experimental study on bond strength of deformed steel bars in recycled glass aggregate concrete. *KSCCE Journal of Civil Engineering*, 22(9), 3409–3418. <https://doi.org/10.1007/s12205-018-0051-5>
- Wang, J. Z., Yang, L., & Yang, J. L. (2020). Bond behavior of epoxy-coated reinforcing bars with seawater sea-sand concrete. *Aci Structural Journal*, 117(4), 193–208. <https://doi.org/10.14359/51723510>
- Wardeh, G., Ghorbel, E., Gomart, H., & Fiorio, B. (2017). Experimental and analytical study of bond behavior between recycled aggregate concrete and steel bars using a pullout test. *Structural Concrete*, 18(5), 811–825. <https://doi.org/10.1002/suco.201600155>
- Wu, Y.-F., & Zhao, X.-M. (2013). Unified bond stress-slip model for reinforced concrete. *Journal of Structural Engineering*, 139(11), 1951–1962. [https://doi.org/10.1061/\(asce\)st.1943-541x.0000747](https://doi.org/10.1061/(asce)st.1943-541x.0000747)
- Xiao, J. Z., & Falkner, H. (2007). Bond behaviour between recycled aggregate concrete and steel rebars. *Construction and Building Materials*, 21(2), 395–401. <https://doi.org/10.1016/j.conbuildmat.2005.08.008>
- Xing, G. H., Zhou, C., Wu, T., & Liu, B. Q. (2015). Experimental study on bond behavior between plain reinforcing bars and concrete. *Advances in Materials Science and Engineering*, 2015, 9. <https://doi.org/10.1155/2015/604280>
- Yang, H., Deng, Z., Qin, Y., & Lv, L. (2015). A study on the bond behavior of corroded reinforced concrete containing recycled aggregates. *Advances in Materials Science and Engineering*, 2015, 1–9. <https://doi.org/10.1155/2015/249301>
- Yeih, W., Huang, R., Chang, J. J., & Yang, C. C. (1997). A pullout test for determining interface properties between rebar and concrete. *Advanced Cement Based Materials*, 5(2), 57–65. [https://doi.org/10.1016/s1065-7355\(96\)00004-1](https://doi.org/10.1016/s1065-7355(96)00004-1)
- Zhang, C., Chun, Q., Jin, H., Zhang, S., & Lin, Y. (2021). Experimental and numerical research on the bond-slip behavior of square-rebar reinforced concrete during the republic of China (1912–1949). *Structural Concrete*, 22(2), 871–893. <https://doi.org/10.1002/suco.202000393>
- Zhang, S., Fan, Y., Jia, Z., & Ren, J. (2021). Effect of nano-kaolin clay on rebar corrosion and bond behavior between rebar and concrete. *Journal of Materials in Civil Engineering*. [https://doi.org/10.1061/\(asce\)mt.1943-5533.0003529](https://doi.org/10.1061/(asce)mt.1943-5533.0003529)
- Zhang, J., He, T., & Suvrit Sra, A. J. (2020b). "Why gradient clipping accelerates training: A theoretical justification for adaptivity." *arXiv e-prints*, arXiv:1905.11881v11882.
- Zhang, B., Zhu, H., Chen, J., & Yang, O. (2020). Influence of specimen dimensions and reinforcement corrosion on bond performance of steel bars in concrete. *Advances in Structural Engineering*, 23(9), 1759–1771. <https://doi.org/10.1177/1369433219900682>
- Zhou, Q. S., Lu, C. F., Wang, W., Wei, S. H., & Xi, B. F. (2020). Effect of fly ash and corrosion on bond behavior in reinforced concrete. *Structural Concrete*, 21(5), 1839–1852. <https://doi.org/10.1002/suco.201900264>
- Zhou, Z. D., & Qiao, P. Z. (2018). Bond behavior of epoxy-coated rebar in ultra-high performance concrete. *Construction and Building Materials*, 182, 406–417. <https://doi.org/10.1016/j.conbuildmat.2018.06.113>
- Zhou, Y., Zheng, S., Huang, Z., Sui, L., & Chen, Y. (2020). Explicit neural network model for predicting FRP-concrete interfacial bond strength based on a large database. *Composite Structures*. <https://doi.org/10.1016/j.comps.2020.111998>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
