

# Adaptive Robust Stochastic Configuration Networks for Near-Infrared Multivariate Analysis

Yuqiang Li<sup>✉</sup>, Wenli Du<sup>✉</sup>, Senior Member, IEEE, Xinjie Wang<sup>✉</sup>, Minglei Yang<sup>✉</sup>, and Yunmeng Zhao<sup>✉</sup>

**Abstract**—Near-infrared (NIR) technology has gained wide acceptance in practical processes and is now the measurement of choice in many sectors. However, with increasing spectral dimensionality, it is challenging to establish a prediction model with satisfactory stability and generalization. Stochastic configuration networks (SCNs) based on supervisory learning mechanism have demonstrated significant advantages in developing nonlinear learners. However, existing incremental learning strategies make it difficult to achieve fast convergence while obtaining a suitable-scale network in high-dimensional spectra modeling. In addition, the linear or regularization weight estimation methods are vulnerable to outliers and noise in NIR analysis. To accelerate model construction and improve model performance in high-dimensional spectra analysis, the adaptive robust SCN (AR-SCN) algorithm is proposed in this work, which can perform adaptive incremental learning according to the prediction residual and robustly estimate the output weights by the global-local shrinkage strategy. Comparison results on three benchmark NIR datasets and real-world gasoline blending process verify the effectiveness of the proposed method. Compared with the state-of-the-art SCNs, the AR-SCN method can simultaneously improve the construction efficiency and robustness of SCNs.

**Index Terms**—Adaptive incremental learning, multivariate modeling, near-infrared (NIR) spectroscopy, sparse Bayesian learning (SBL), stochastic configuration networks (SCNs).

## I. INTRODUCTION

NEAR-INFRARED (NIR) analytical methods provide valuable information about the composition and structure of matter through electromagnetic radiation and have been widely used to understand the properties and behavior of substances in many fields [1], [2], [3]. As a data-driven approach,

Received 28 April 2024; revised 31 August 2024; accepted 29 November 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3305900, in part by the National Natural Science Foundation of China (Basic Science Center Program) under Grant 61988101, in part by the Major Program of Qingyuan Innovation Laboratory under Grant 00122002, and in part by the Fundamental Research Funds for the Central Universities. (*Corresponding author: Wenli Du*)

Yuqiang Li, Minglei Yang, and Yunmeng Zhao are with the Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: yuqiangli@mail.ecust.edu.cn; mlyang@ecust.edu.cn; yunmeng.zhao@ecust.edu.cn).

Wenli Du is with the Key Laboratory of Smart Manufacturing in Energy Chemical Process and the Engineering Research Center of Process System Engineering, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China, and also with Qingyuan Innovation Laboratory, Quanzhou 362801, China (e-mail: wldu@ecust.edu.cn).

Xinjie Wang is with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China (e-mail: wangxinjie0621@foxmail.com).

Digital Object Identifier 10.1109/TNNLS.2024.3512492

appropriate chemometric methods are required to construct potential relationships between spectral distributions or intensities and fundamental properties [4]. However, most existing methods make unstable and unreliable prediction in the NIR calibration analysis. For NIR multivariate analysis, the spectra collected from actual complex processes usually contain interferences such as instrument vibration, ambient light, and fluorescence interference, which inevitably affect the extraction of the spectral characteristic information [4]. In addition, due to the contradiction between the development of advanced optical instruments and restricted sampling conditions, the obtained spectra are characterized by high-dimensional small samples, which makes it more challenging to establish models with satisfactory performance [5].

Over the past decades, several methods have been developed for NIR modeling and application analysis. Among the existing methods, linear methods have shown their advantages in terms of construction efficiency and model interpretability [6]. However, changes in the reference materials, sample loading conditions, ambient temperature, and physical state of the particles during data collection may result in nonlinear characteristics of the measured spectra [7]. To this end, a series of nonlinear NIR analysis approaches have been developed, among which neural networks have been widely used because of their universal approximation and nonlinear mapping capabilities, including artificial neural network (ANN) [8], deep autoencoder (DAE) [9], and convolutional neural networks (CNN) [10]. However, the inherent shortcomings of these methods, including slow convergence, local optimum, and gradient explosion, significantly limit their application scope in NIR analysis [11]. Furthermore, limited by sampling cost and data quality, dataset size is one of the possible bottlenecks in developing deep networks in NIR multivariate analysis [12].

To improve the model performance, the incremental learning method that builds the learner by gradually adding network nodes was developed for network construction [13], [14]. Nevertheless, extensive searches for suitable model structures and parameters are time-consuming, especially for large-scale tasks [15]. As an alternative, random learning methods have shown promising advantages in rapidly building neural networks, where the weights and biases between the input and hidden layers are randomly assigned from a set range, and the output weights are obtained by linear solving or regularization methods [16], [17]. In [18] and [19], the random vector functional-link (RVFL) network was proposed for variable

projection analysis with multivariate inputs and performed well in practice. However, generating hidden layer parameters at a constant interval does not guarantee the general approximation capability of the resulting learner [20], [21]. To ensure the universal approximation of the stochastic model, Wang and Li [22] proposed the stochastic configuration networks (SCNs) algorithm based on the random basis function approximation theory, in which the weights and biases of the input layer are produced in varying ranges, and the hidden nodes are added using the performance-based inequality supervisory mechanism. Compared with RVFL, SCNs can obtain a learner with general mapping capabilities while ensuring convergence and showing satisfactory performance in various fields [23], [24], [25]. To improve the learning efficiency of SCNs, Dai et al. [26] developed the block-incremental SCN (BSCN) by adding nodes in batches during the learning process. Although the appropriate setting for the incremental blocks can reduce training time, it is difficult to determine a suitable model structure with a fixed incremental block size. In [27], the hybrid parallel SCN combining point increment and block increment strategies is proposed to improve learning efficiency, but the parallel learning strategy doubles the number of hidden layer parameters and easily leads to model redundancy. Recently, to improve the generalization performance of SCN, the online self-learning SCN that dynamically adjusts the parameter and structure of the learner is developed for nonstationary data [28]. In addition, the regularized SCN is proposed to overcome the uncertainties and multicollinearity in the measured data [29]. However, the learning efficiency of the above methods is still limited by point incremental learning.

For high-dimensional small-sample NIR datasets, the complexity of the neural network architecture will increase as the dimensionality of the input spectra increases, leading to the ill-posed problem in the pseudoinverse parameter estimation and ultimately reducing the generalization ability of the built SCNs in NIR multivariate analysis [30]. To reduce the model complexity and improve the generalizability of the obtained learner in high-dimensional NIR analysis, it is feasible to simplify the model structure and avoid the ill-posed problem by introducing appropriate regularization penalties [31], [32]. However, the inconsistent sparsification of  $L_1$ -norm and the nonsparsity shrinkage of  $L_2$ -norm limit their effectiveness in model simplification [33], [34]. Recently, the Bayesian inference method has been extensively studied for model sparsity, in which the sparse distribution is assumed for model parameters [35], [36], [37]. Compared to existing methods, Bayesian inference has advantages in incorporating prior information on parameter distributions, representing uncertainty in prediction results, and autonomously learning model parameters from existing data [35], [36]. In the past few years, series distributions have been used for sparse parameter learning, such as Gaussian [38], Laplacian [39], and Student's t [40]. However, these methods are limited by the lack of adaptive sparsity and tail robustness, which makes it challenging to achieve overall sparsity while avoiding overshrinkage of important parameters, especially for high-dimensional spectra [41], [42], [43].

For multivariate analysis of the high-dimensional NIR spectra, fast training in the initial stage and precise fine-tuning in the low convergence stage is a proper solution to accelerate the construction process and obtain small-scale network architectures. Meanwhile, sparse learning of the hidden layer output weights is of great significance for improving the robustness of the resulting NIR model. Therefore, adaptively adjusting the size of the incremental blocks according to the model performance and robustly estimating the model weights is a feasible way to obtain the small-scale network with generalization and robustness in NIR multivariate analysis. To this end, the adaptive robust SCN (AR-SCN) is proposed for NIR modeling in this work, which can simultaneously improve the construction efficiency and robustness of SCNs. Comparative results on three benchmark NIR datasets and the gasoline blending process verified the superiority of the proposed AR-SCN method in NIR analysis. The main contributions of this work are summarized as follows.

- 1) To accelerate the SCNs construction process and obtain a suitable network structure in high-dimensional spectra modeling, an adaptive incremental learning strategy is proposed, which allows the incremental block size to be adaptively adjusted according to the model residual.
- 2) To improve the robustness of SCNs in NIR prediction, the global-local shrinkage mechanism is used to shrink the globally irrelevant weights while retaining locally relevant weights. In addition, the posterior distributions of the model parameters are derived.
- 3) Comparison experiments conducted on benchmark NIR datasets and real gasoline blending process NIR show that the AR-SCN method has better prediction accuracy and robustness than existing methods in the NIR analysis.

The rest of this article is organized as follows. Related works of SCNs and Bayesian inference framework are briefly introduced in Section II. The proposed method is presented in detail in Section III. Section IV presents the experimental datasets and discusses the parameter setting. Section V reports the prediction results. Finally, the conclusion and future works are given in Section VI.

## II. RELATED WORKS

This section briefly introduces the SCN algorithm principles and different increment strategies for model structure learning. In addition, the Bayesian inference for sparse parameter estimation has been discussed.

### A. Stochastic Configuration Networks

SCNs are a novel randomized method that builds a single hidden layer feedforward neural network in a constructive approach. Compared with traditional learning methods, the weights and biases between input and hidden layers are randomly produced in varying parameter ranges  $[-\vartheta_k, \vartheta_k]$  related to iteration times  $k$ , and the hidden nodes are determined under inequality supervisory mechanism [22].

Given a dataset  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ , real-valued input  $\mathbf{x}_i = \{x_{i,1}, \dots, x_{i,d}\} \in \mathbb{R}^d$ , and corresponding response

$\mathbf{y}_i = \{y_{i,1}, \dots, y_{i,m}\} \in \mathbb{R}^m$  with  $i = 1, 2, \dots, N$ ,  $N$  denotes instance count, and  $d$  and  $m$  present the dimensions of  $\mathbf{x}_i$  and  $\mathbf{y}_i$ , respectively. Given a target function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , the SCN model  $f_{L-1}$  with  $L - 1$  hidden nodes can be expressed as

$$f_{L-1}(\mathbf{x}) = \sum_{j=1}^{L-1} \beta_j g_j(\mathbf{w}_j^\top \cdot \mathbf{x} + b_j) = \mathbf{h}(\mathbf{x})\beta \quad (1)$$

where  $\beta = [\beta_1, \beta_2, \dots, \beta_{L-1}]^\top$  denotes the output weights;  $g_j(\cdot)$  denotes the random basis function;  $\mathbf{w}_j \in \mathbb{R}^d$  and  $b_j \in \mathbb{R}$  denote input weights and bias between input layer and the  $j$ th hidden node, respectively;  $\mathbf{h}(\mathbf{x}) = [g_1(\mathbf{x}), \dots, g_{L-1}(\mathbf{x})]$  denotes the hidden layer output vector; and the superscript  $\top$  denotes transpose.

The objective of SCNs is to find  $f_{L-1}$  to minimize the prediction error  $\|f - f_{L-1}\|^2$  with the least hidden layer width and learning time. In the incremental learning framework, the new nodes are gradually added to the current learner until the termination condition is met. For point incremental learning, the additive model of the existing network and new nodes can be expressed as follows:

$$f_L(\mathbf{x}) = f_{L-1}(\mathbf{x}) + g_L(\mathbf{x})\beta_L \quad (2)$$

where

$$\beta_L = \frac{\langle \mathbf{e}_{L-1}, g_L \rangle}{\|g_L\|^2} \quad (3)$$

is the output weight of the incremental evaluation and  $\mathbf{e}_{L-1} = f - f_{L-1}$  is the prediction residual of the previous learner  $f_{L-1}$ .

Three SCN algorithms with different weight updating strategies, namely, point updating, window updating, and global updating, are proposed to estimate the hidden layer output weight [22]. From the optimization perspective, the global updating method usually outperforms the others, mainly attributed to its global least-squares updating strategy. Therefore, the global update method is denoted as SCN in the rest of this article. More details about the SCN can be found in [22].

Different from the SCN based on point incremental learning, the BSCN method based on block-incremental learning allocates nodes in batches in each iteration as follows [26]:

$$f_{L^{(k)}}(\mathbf{x}) = f_{L^{(k-1)}}(\mathbf{x}) + \mathbf{h}_{\Delta_k}(\mathbf{x})\beta_{\Delta_k} \quad (4)$$

where  $L^{(k)}$  denotes the number of hidden layer node after the  $k$ th iteration and  $\Delta_k$  denotes the added node block in the  $k$ th iteration, which is fixed in the BSCN learning process.  $\mathbf{h}_{\Delta_k}(\mathbf{x}) = [g_{L^{(k-1)}+1}(\mathbf{x}), \dots, g_{L^{(k)}}(\mathbf{x})]_{\Delta_k}$  represents the hidden layer output block vector in the  $k$ th increment learning, and  $\beta_{\Delta_k} = [\beta_{L^{(k-1)}+1}, \dots, \beta_{L^{(k)}}]_{\Delta_k}^\top$  is the output weight block.

### B. Bayesian Inference

For the given dataset  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ , the prediction output of the SCN satisfying the preset conditions can be expressed as

$$\mathbf{y} = \mathbf{H}\beta + \epsilon \quad (5)$$

where  $\mathbf{H} = [\mathbf{h}^\top(\mathbf{x}_1), \dots, \mathbf{h}^\top(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times L}$  represents the hidden layer output matrix and  $\epsilon$  is an  $N$ -dimensional noise

vector. Due to the dependence on the number of observations and variables to be estimated, the output weights obtained by ordinary least squares often have a low bias but large variance [44]. To improve prediction accuracy, the Bayesian inference method has been applied for sparse weight learning, where target weights with sparsity are inferred in the form of probability distributions by introducing reasonable prior knowledge. For (5), the likelihood distribution of the SCN can be written as

$$p(\mathbf{y} | \mathbf{x}, \beta) = \prod_{i=1}^N p(y_i | \mathbf{h}(\mathbf{x}_i), \beta). \quad (6)$$

Based on the prior and likelihood beliefs on the weight distribution, the posterior distribution can be obtained as follows:

$$p(\beta | \mathbf{x}, \mathbf{y}) = \frac{1}{Z} p(\mathbf{y} | \mathbf{H}, \beta) p(\beta) \quad (7)$$

where  $Z$  is a normalization constant and is independent of  $\beta$ . For a new input  $\hat{\mathbf{x}}$ , the predictive distribution of the response variable can be expressed as follows:

$$p(\hat{y} | \mathbf{x}, \mathbf{y}; \hat{\mathbf{x}}) = \int p(\hat{y} | \hat{\mathbf{x}}, \beta) p(\beta | \mathbf{x}, \mathbf{y}) d\beta. \quad (8)$$

### III. PROPOSED AR-SCN METHOD

A novel SCN-based method is proposed in this section to accelerate the model construction and improve the model performance in NIR multivariate analysis. First, the measured NIR datasets [see Fig. 1(a)] are used as the input of the AR-SCN [see Fig. 1(b)]. The proposed AR-SCN consists of two main parts, namely, adaptive incremental learning [see Fig. 1(c)] and robust weight estimation [see Fig. 1(d)], which are utilized to accelerate the construction process and determine the output weights of the constructed model, respectively. Finally, the obtained AR-SCN model can be used for NIR prediction analysis [see Fig. 1(e)].

#### A. Adaptive Incremental Learning SCN

The SCN with point increment can build small or compact predictive models with slow learning speed. In contrast, the BSCN with block increment can significantly reduce the computational cost, but it is prone to overfitting. For high-dimensional NIR analysis, fast training in the initial stage and precise fine-tuning in the low convergence stage is a suitable solution to accelerate the construction process and avoid model redundancy. To balance the model complexity and construction efficiency of the SCN in NIR multivariate analysis, an adaptive incremental learning strategy is proposed in Theorem 1, which can adaptively adjust the incremental block size according to the model residual during the learning process.

For the SCN model  $f_{L^{(k)}}$  with  $L^{(k)}$  nodes after the  $k$ th iteration, let  $\beta^* = \arg \min_{\beta} \|f - \sum_{j=1}^{L^{(k)}} g_j(\mathbf{x})\beta_j\|$ ,  $\mathbf{e}_{L^{(k)}-\Delta_k}^* = f - \mathbf{H}_{L^{(k-1)}}\beta_{L^{(k-1)}}^*$  denote the  $k$ th iteration residual block under global least squares, and define intermediate values  $\tilde{\mathbf{e}}_{L^{(k)}} = \mathbf{e}_{L^{(k)}-\Delta_k}^* - \mathbf{H}_{\Delta_k}\tilde{\beta}_{\Delta_k}$  and  $\tilde{\beta}_{\Delta_k} = (\mathbf{H}_{\Delta_k}^\top \mathbf{H}_{\Delta_k})^\dagger \mathbf{H}_{\Delta_k}^\top \mathbf{e}_{L^{(k)}-\Delta_k}^*$ , where  $\mathbf{H}_{\Delta_k} = [\mathbf{h}_{\Delta_k}^\top(\mathbf{x}_1), \dots, \mathbf{h}_{\Delta_k}^\top(\mathbf{x}_N)]^\top$  denotes the hidden layer

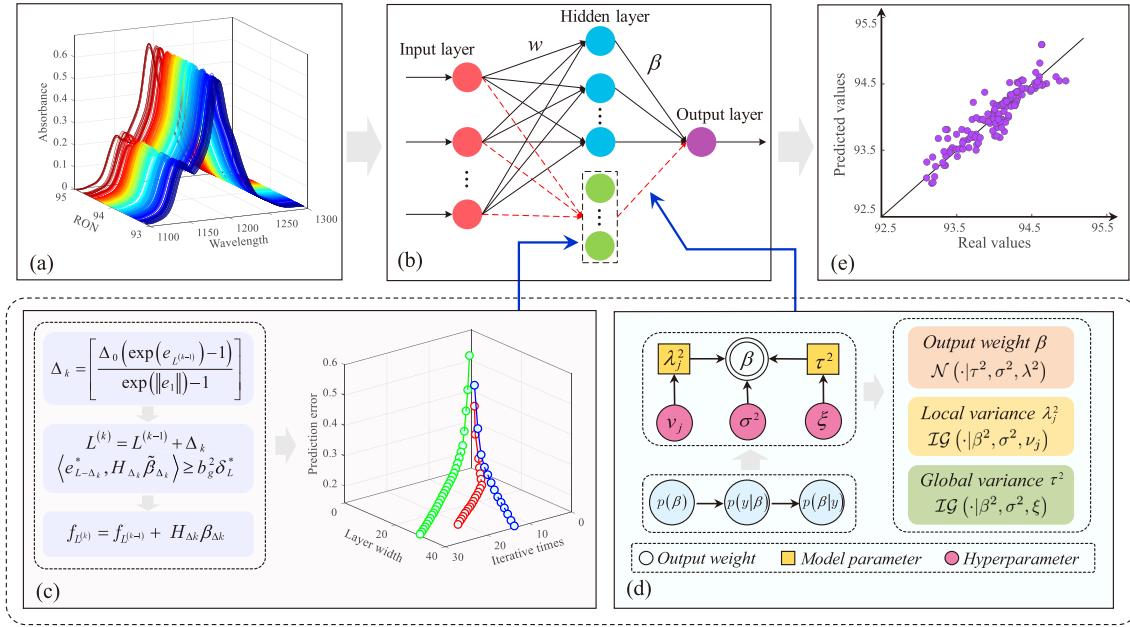


Fig. 1. Schematic of the AR-SCN method. (a) Measured dataset. (b) AR-SCN. (c) Adaptive incremental learning. (d) Robust weight estimation. (e) Prediction analysis.

output block matrix in the  $k$ th increment learning and  $\dagger$  denotes the Moore–Penrose generalized inverse and  $\mathbf{e}_0^* = f$ .

**Theorem 1:** Let  $\Gamma := \{g_1, g_2, g_3, \dots\}$  be a set of real-valued functions, and  $\text{span}(\Gamma)$  denotes a function space spanned by  $\Gamma$ . Suppose that  $\text{span}(\Gamma)$  is dense in the space of all Lebesgue measurable functions and  $\forall g \in \Gamma, 0 < \|g\| < b_g$  for some positive real numbers  $b_g \in \mathbb{R}^+$ . Given learning parameter  $r \in (0, 1)$  and a nonnegative real number sequence  $\{\mu_L\}$ , with  $\lim_{L^{(k)} \rightarrow +\infty} \mu_{L^{(k)}} = 0$  and  $0 \leq \mu_{L^{(k)}} \leq (1-r)$  for  $k = 1, 2, \dots$ , for  $L^{(k)} = 1, 2, \dots, \Delta_k \in \{L\}$ , the constraint coefficient can be denoted as

$$\eta_{L^{(k)},q}^* = (1 - r - \mu_{L^{(k)}}) \|\mathbf{e}_{L^{(k)}-\Delta_k,q}^*\|^2, q = 1, 2, \dots, m. \quad (9)$$

The increment block  $\Delta_k$  is adaptively determined as follows:

$$\Delta_k = \left\lceil \frac{\Delta_0 (\exp(\|\mathbf{e}_{L^{(k)-1}}\|^2) - 1)}{\exp(\|\mathbf{e}_1\|^2) - 1} \right\rceil \quad (10)$$

where  $\Delta_0$  denotes the initial block,  $\|\mathbf{e}_1\| = \|f - f_1(\mathbf{x})\|$ , and  $\Delta_k$  can be adaptively adjusted from the scope  $[1, \Delta_0]$  according to the previous prediction residual  $\|\mathbf{e}_{L^{(k)-1}}\| = \|f - f_{L^{(k)-1}}(\mathbf{x})\|$ . Then, if the hidden layer output block  $\mathbf{H}_{\Delta_k}$  is determined by the following supervisory mechanism:

$$\langle \mathbf{e}_{L^{(k)}-\Delta_k,q}^*, \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q} \rangle \geq b_g^2 \eta_{L^{(k)},q}^*, q = 1, 2, \dots, m \quad (11)$$

the width of the hidden layer gradually increases during the iteration process

$$L^{(k)} = L^{(k-1)} + \Delta_k \quad (12)$$

and the output weights are calculated by

$$\{\boldsymbol{\beta}_1^*, \dots, \boldsymbol{\beta}_{L^{(k)}}^*\} = \arg \min_{\boldsymbol{\beta}} \|f - \sum_{j=1}^{L^{(k)}} g_j(\mathbf{x}) \boldsymbol{\beta}_j\|. \quad (13)$$

Then, we have  $\lim_{L^{(k)} \rightarrow +\infty} \|f - f_{L^{(k)}}^*\| = 0$ .

*Proof:* For the residual sequence  $\|\mathbf{e}_{L^{(k)}}^*\|^2$ , we have

$$\begin{aligned} \|\mathbf{e}_{L^{(k)}}^*\|^2 &\leq \|\tilde{\mathbf{e}}_{L^{(k)}}\|^2 \\ &= \langle \mathbf{e}_{L^{(k)}-\Delta_k}^* - \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k}, \mathbf{e}_{L^{(k)}-\Delta_k}^* - \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k} \rangle \\ &= \|\mathbf{e}_{L^{(k)}-\Delta_k}^*\|^2 - \|\mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k}\|^2 \\ &\leq \|\mathbf{e}_{L^{(k)}-\Delta_k}^*\|^2. \end{aligned} \quad (14)$$

Combined with the supervisory mechanism (11), we can easily get that  $\|\mathbf{e}_{L^{(k)}}^*\|^2 \leq \|\tilde{\mathbf{e}}_{L^{(k)}}\|^2 \leq \|\mathbf{e}_{L^{(k)}-\Delta_k}^*\|^2 \leq \|\tilde{\mathbf{e}}_{L^{(k)}-\Delta_k}\|^2$ , and thus,  $\|\mathbf{e}_{L^{(k)}}^*\|^2$  is monotonically decreasing. From (12) and (14), we can further obtain

$$\begin{aligned} &\|\mathbf{e}_{L^{(k)}}^*\|^2 - (r + \mu_{L^{(k)}}) \|\mathbf{e}_{L^{(k)}-\Delta_k}^*\|^2 \\ &\leq \|\tilde{\mathbf{e}}_{L^{(k)}}\|^2 - (r + \mu_{L^{(k)}}) \|\mathbf{e}_{L^{(k)}-\Delta_k}^*\|^2 \\ &= \sum_{q=1}^m \left( \langle \mathbf{e}_{L^{(k)}-\Delta_k,q}^* - \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q}, \mathbf{e}_{L^{(k)}-\Delta_k,q}^* - \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q} \rangle \right. \\ &\quad \left. - (r + \mu_{L^{(k)}}) \langle \mathbf{e}_{L^{(k)}-\Delta_k,q}^*, \mathbf{e}_{L^{(k)}-\Delta_k,q}^* \rangle \right) \\ &= \sum_{q=1}^m \left( (1 - r - \mu_{L^{(k)}}) \langle \mathbf{e}_{L^{(k)}-\Delta_k,q}^*, \mathbf{e}_{L^{(k)}-\Delta_k,q}^* \rangle \right. \\ &\quad \left. - 2 \langle \mathbf{e}_{L^{(k)}-\Delta_k,q}^*, \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q} \rangle + \langle \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q}, \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q} \rangle \right) \\ &= \sum_{q=1}^m \left( \eta_{L^{(k)},q}^* - \langle \mathbf{e}_{L^{(k)}-\Delta_k,q}^*, \mathbf{H}_{\Delta_k} \tilde{\boldsymbol{\beta}}_{\Delta_k,q} \rangle \right) \leq 0. \end{aligned} \quad (15)$$

Therefore, we have the following inequality:

$$\|\mathbf{e}_{L^{(k)}}^*\|^2 \leq (r + \mu_{L^{(k)}}) \|\mathbf{e}_{L^{(k)}-\Delta_k}^*\|^2. \quad (16)$$

**Algorithm 1** SCN With Adaptive Incremental Learning

**Input:** Dataset  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ , initial block size  $\Delta_0$ ;  
**Output:** Hidden layer output matrix  $\mathbf{H}$ ;

- 1: Initialize: maximum number of hidden nodes  $P$ , maximum iteration times  $K$ , tolerance error  $\delta$ , and set  $k = 0$ ;
- 2: **While**  $k \leq K$  and  $\|\mathbf{e}_f\| > \delta$ , **do**
- 3: Generate the random weight  $\mathbf{w}_j$  and bias  $b_j$ ;
- 4: Obtain the candidate nodes from (9);
- 5: Add the new block nodes  $\Delta_k$  using (10);
- 6: Update the hidden layer width  $L$  using (12);
- 7: Calculate the output weight  $\beta^*$  by (13);
- 8: Renew  $k \leftarrow k + 1$ ,  $\mathbf{e}_f = f - f_{L(k)}^*$ ;
- 9: **End While**
- 10: Output hidden ouput matrix  $\mathbf{H}$  of the built SCN.

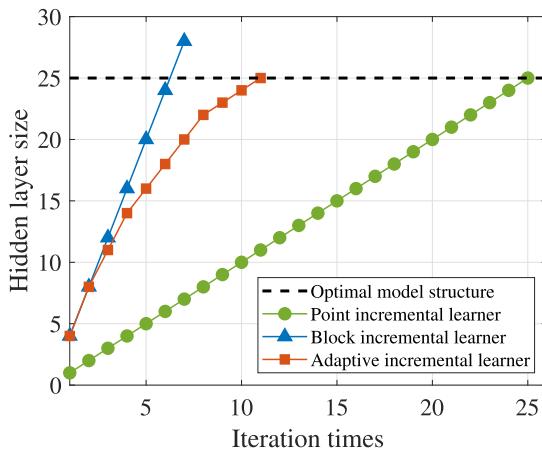


Fig. 2. Model width differences of different incremental learning algorithms.

Note that  $\lim_{L(k) \rightarrow +\infty} \mu_{L(k)} \|\mathbf{e}_{L(k)-\Delta_k}^*\|^2 = 0$  when  $\lim_{L(k) \rightarrow +\infty} \mu_{L(k)} = 0$ . Based on (14), it is easily verified that  $\lim_{L(k) \rightarrow +\infty} \|\mathbf{e}_{L(k)}^*\|^2 = 0$ , that is,  $\lim_{L(k) \rightarrow +\infty} \|\mathbf{e}_{L(k)}^*\| = 0$ . ■

Based on the above description, the proposed adaptive incremental learning method is summarized in Algorithm 1.

*Remark 1:* Fig. 2 depicts an example to compare the differences between existing incremental learning algorithms on the same task, where the model prediction residual decays exponentially. The initial block size  $\Delta_0$  of the block-incremental learner and the adaptive incremental learner is set to 4, and the dashed line is the hidden layer width of the model with satisfactory performance. It can be found that the block-incremental learner reduces iterations by adding hidden layer nodes in batches but suffers from overfitting. As indicated by the green line, the point incremental learner determines the small-scale model through multiple iterations of point-by-point increments. Through adaptive incremental block adjustment based on prediction residual, the proposed adaptive incremental learner can reduce the iteration times through block increment in the initial stage and determine the appropriate model through point increment in the final stage, thereby accelerating the model construction process and avoiding model redundancy.

**B. Robust Estimation of Output Weights**

For the given dataset  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ , the probability distribution of the SCN constructed by the proposed adaptive incremental learning strategy can be represented as

$$p(\mathbf{y} | X; \boldsymbol{\beta}, \sigma^2) = \mathcal{N}(\mathbf{H}\boldsymbol{\beta}, \sigma^2) \quad (17)$$

where  $\sigma^2$  denotes the variance of the noise. In the sparse Bayesian learning (SBL) framework, it is generally assumed that the model output weight  $\boldsymbol{\beta}$  follows some prior distributions. However, as the probability density of different priors shown in Fig. 3(a), the limitation of commonly used distributions, such as Laplacian, Cauchy, and Strawderman–Berger, is that uniform shrinkage is performed to all the weights, which makes it easier to promote overall sparsity by overshrinking large coefficients with significant explanatory power [42].

The horseshoe prior is a continuous distribution characterized by sharp peak and heavy tails, as shown in Fig. 3(a). Distinguished from traditional priors that impose a constant amount of shrinkage on all coefficients regardless of their magnitude, the horseshoe prior provides collaborative shrinkage by combining the local and global shrinkage, enabling it to prioritize locally pertinent weights while filtering out global extraneous components. Specifically, for the weight  $\beta_j$ , the shrinkage mechanism of the horseshoe prior can be expressed as follows:

$$\begin{aligned} p(\boldsymbol{\beta}_j | \lambda_j^2, \tau^2, \sigma^2) &= \mathcal{N}(0, \lambda_j^2 \tau^2 \sigma^2) \\ p(\lambda_j | \tau) &= \mathcal{C}(0, \tau) \\ p(\tau^2 | \sigma^2) &= \mathcal{C}(0, \sigma^2) \end{aligned} \quad (18)$$

where  $\{\lambda_j\}_{j=1}^L$  is the local shrinkage parameter,  $\tau$  is the global shrinkage parameter,  $\mathcal{C}(\cdot)$  represents the Cauchy distribution, and the variance  $\sigma^2$  follows Jeffreys' prior, i.e.,  $p(\sigma^2) = J(\sigma^2)$ . For the model weight  $\boldsymbol{\beta}$ , the local shrinkage parameters  $\{\lambda_j\}_{j=1}^L$  allow the model to prioritize the locally relevant weights. Therefore, for weights associated with significant effects, the local shrinkage parameter  $\lambda_j$  can shrink less to retain these weights [45], [46]. For irrelevant weights, smaller local shrinkage parameter  $\lambda_j$  can shrink these parameters to zero. Meanwhile, the overall sparsity promoted by the global shrinkage parameter  $\tau$  can effectively filter out globally irrelevant weights [42]. Combining the global shrinkage factor  $\tau$  with the local shrinkage parameters  $\{\lambda_j\}_{j=1}^L$  allows for selective shrinkage of the regression weight  $\boldsymbol{\beta}$ , effectively distinguishing between significant weights and irrelevant weights in NIR modeling. In addition, the collaborative shrinkage of the horseshoe distribution can adaptively determine the appropriate shrinkage level of weights through Bayesian inference.

As a member of multivariate-normal scale mixtures, the differences in the suitability of the horseshoe prior from other distributions can be further illustrated by representing the shrinkage profiles [45]. Assume that  $\tau^2 = \sigma^2 = 1$  and define  $\kappa_j = 1/(1 + \lambda_j^2)$ , and the estimator for  $\boldsymbol{\beta}_j$  can be defined as

$$\begin{aligned} \mathbb{E}(\boldsymbol{\beta}_j | \mathbf{y}, \lambda_j^2) &= \left(1 - \frac{1}{1 + \lambda_j^2}\right) \mathbf{y}_j + \left(1 - \frac{1}{1 + \lambda_j^2}\right) \mathbf{0} \\ &= (1 - \kappa_j) \mathbf{y}_j. \end{aligned} \quad (19)$$

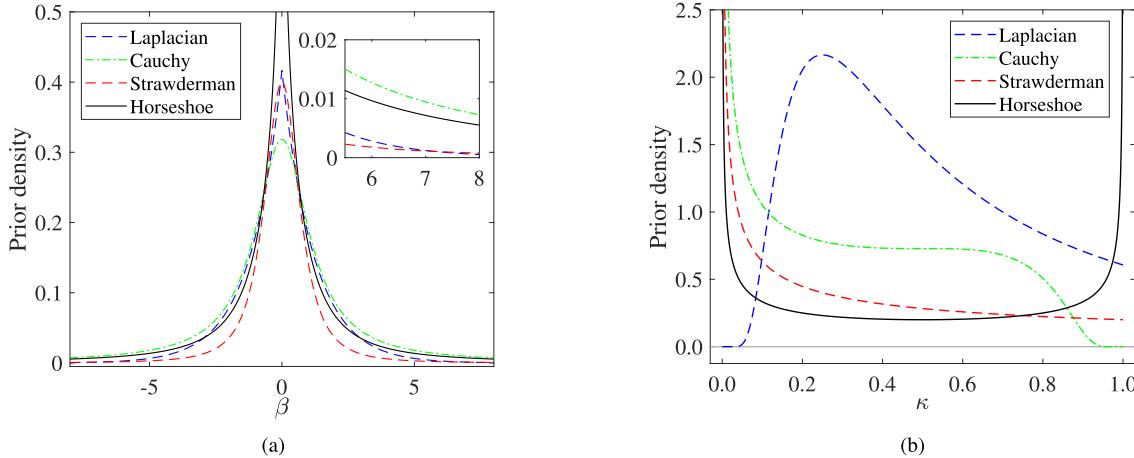


Fig. 3. Comparison of the densities for (a)  $p(\beta)$  and (b)  $p(\kappa)$ , where  $\kappa = 0$  denotes no shrinkage and  $\kappa = 1$  represents total shrinkage.

Based on Fubini's theorem, we have

$$\begin{aligned} \mathbb{E}(\beta_j | \mathbf{y}) &= \int_0^1 (1 - \kappa_j) y_j p(\kappa_j | \mathbf{y}) d\kappa_j \\ &= \{1 - \mathbb{E}(\kappa_j | \mathbf{y})\} y_j. \end{aligned} \quad (20)$$

Then, the shrinkage coefficient  $\kappa_j \in [0, 1]$  provides an intuitive way to understand how the model distinguishes significant weights through global-local shrinkage. As similar priors for  $\kappa$  shown in Fig. 3(b), the shrinkage profile of the horseshoe prior is unbounded at both  $\kappa_j = 0$  and  $\kappa_j = 1$ , indicating that it allows significant weights to remain large ( $\kappa_j \approx 0$ , no shrinkage) and pulls irrelevant weights close to zero ( $\kappa_j \approx 1$ , strong shrinkage) [46]. Meanwhile, the overall degree of sparsity in  $\beta$  is controlled by the global shrinkage parameter  $\tau$  by placing one-third of density mass on  $1/4 \leq \kappa_j \leq 3/4$ , allowing moderate shrinkage of each  $\beta_j$ . In contrast, no other commonly used priors have both of these characteristics. Specifically, the shrinkage profile of the Laplacian prior disappears near  $\kappa = 0$ , which means that the large weights are overshrunk. At the same time, the Laplacian, Cauchy, and Strawderman-Berger priors all tend to be fixed constants near  $\kappa = 1$ , limiting the capability of these estimators to suppress sparse weights to zero.

In this work, the horseshoe prior is adopted to improve the robustness of the SCN-based NIR model. To obtain the posterior distribution of the model parameters, the auxiliary variables  $v_j$  ( $j = 1, \dots, L$ ) and  $\xi$  are introduced, and we have

$$p(\lambda_j^2 | v_j) = \text{IG}\left(\frac{1}{2}, \frac{1}{v_j}\right) \quad (21)$$

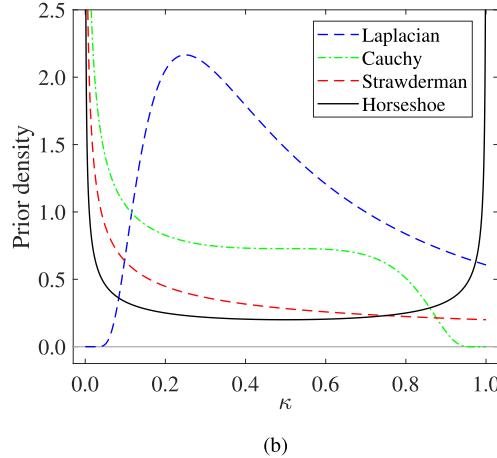
and

$$p(\tau^2 | \xi) = \text{IG}\left(\frac{1}{2}, \frac{1}{\xi}\right) \quad (22)$$

where  $p(v_1, \dots, v_L, \xi) = \text{IG}(1/2, 1)$  and  $\text{IG}(\cdot)$  denotes the inverse-gamma distribution. The parameter posterior distribution can be obtained according to the Bayesian inference.

For the weight coefficient  $\beta$ , the conditional posterior distribution can be derived as follows:

$$p(\beta | \mathbf{y}, \mathbf{X}; \sigma^2)$$



$$\begin{aligned} &\propto p(\mathbf{y} | \mathbf{X}; \beta, \sigma^2) p(\beta | \lambda^2, \tau^2, \sigma^2) \\ &\propto (2\pi\sigma^2)^{-\frac{N+L}{2}} |\Lambda_*|^{-\frac{1}{2}} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\beta\|^2}{2\sigma^2} - \frac{\beta^T \beta}{2\sigma^2 \Lambda_*}\right) \\ &= \mathcal{N}(\mathbf{D}^{-1} \mathbf{H}^T \mathbf{y}, \sigma^2 \mathbf{D}^{-1}) \end{aligned} \quad (23)$$

where  $\mathbf{D} = \mathbf{H}^T \mathbf{H} + \Lambda_*^{-1}$ ,  $\Lambda_* = \tau^2 \Lambda$ , and  $\Lambda = \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_L^2)$ .

For the noise variance  $\sigma^2$ , the conditional posterior distribution is

$$\begin{aligned} &p(\sigma^2 | \mathbf{y}, \mathbf{X}; \beta) \\ &\propto p(\mathbf{y} | \mathbf{X}; \beta, \sigma^2) p(\beta | \lambda^2, \tau^2, \sigma^2) p(\sigma^2) \\ &\propto (\sigma^2)^{-(\frac{N+L}{2}+1)} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\beta\|^2 + \beta^T \Lambda_*^{-1} \beta}{2\sigma^2}\right) \\ &= \mathcal{IG}\left(\frac{N+L}{2}, \frac{\|\mathbf{y} - \mathbf{H}\beta\|^2 + \beta^T \Lambda_*^{-1} \beta}{2}\right). \end{aligned} \quad (24)$$

Following the fast Bayesian approach in [48], the convergence of the sampler can be improved by integrating out the weight coefficients  $\beta$  from the posterior distribution of  $\sigma^2$ . After integration, the conditional posterior density of  $\sigma^2$  is

$$p(\sigma^2 | \mathbf{y}, \mathbf{X}; \beta) = \mathcal{IG}\left(\frac{N}{2}, \frac{\mathbf{y}^T (\mathbf{I}_N - \mathbf{H} \Lambda_*^{-1} \mathbf{H}^T) \mathbf{y}}{2}\right) \quad (25)$$

where  $\mathbf{I}_N \in \mathbb{R}^{N \times N}$  is the identity matrix.

For the local hypervariance  $\lambda_j^2$ , it can be calculated as

$$\begin{aligned} p(\lambda_j^2 | \beta_j) &\propto p(\beta_j | \lambda_j^2, \tau^2, \sigma^2) p(\lambda_j^2) \\ &\propto (\lambda_j^2)^{-2} \exp\left(-\frac{1}{\lambda_j^2} \left(\frac{\beta_j^2}{2\tau^2 \sigma^2} + \frac{1}{v_j}\right)\right) \\ &= \mathcal{IG}\left(1, \frac{\beta_j^2}{2\tau^2 \sigma^2} + \frac{1}{v_j}\right), j = 1, 2, \dots, L. \end{aligned} \quad (26)$$

Similarly, the conditional posterior distribution for the global hypervariances  $\tau^2$  is

$$p(\tau^2 | \beta) \propto p(\beta | \lambda^2, \tau^2, \sigma^2) p(\tau^2)$$

$$\begin{aligned} & \propto (\tau^2)^{-2} \exp\left(-\frac{1}{\tau^2}\left(\sum_{j=1}^L \frac{\beta_j^2}{2\sigma^2\lambda_j^2} + \frac{1}{\xi}\right)\right) \\ & = \mathcal{IG}\left(1, \sum_{j=1}^L \frac{\beta_j^2}{2\sigma^2\lambda_j^2} + \frac{1}{\xi}\right). \end{aligned} \quad (27)$$

For the auxiliary variables  $v_j$  and  $\xi$ , the conditional posterior distributions are

$$\begin{aligned} p(v_j | \lambda_j^2) & \propto p(\lambda_j^2 | v_j) p(v_j) \\ & \propto v_j^{-2} \exp\left(-\frac{1}{v_j}\left(1 + \frac{1}{\lambda_j^2}\right)\right) \\ & = \mathcal{IG}\left(1, 1 + \frac{1}{\lambda_j^2}\right), j = 1, 2, \dots, L \end{aligned} \quad (28)$$

and

$$\begin{aligned} p(\xi | \tau^2) & \propto p(\tau^2 | \xi) p(\xi) \\ & \propto \xi^{-2} \exp\left(-\frac{1}{\xi}\left(1 + \frac{1}{\tau^2}\right)\right) \\ & = \mathcal{IG}\left(1, 1 + \frac{1}{\tau^2}\right). \end{aligned} \quad (29)$$

Based on the above conditional posterior, the Gibbs sampling method can be used to obtain the posterior distribution of model parameter. The proposed AR-SCN method is summarized in Algorithm 2.

*Remark 2:* Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm that is used for statistical inference when direct solving is difficult, especially for high-dimensional data. For the AR-SCN method, the desired prior distribution over the variables can be obtained by successive Gibbs sampling, in which the number of Gibbs sampling times ( $N_s$ ) and burn-in (bi) time are important parameters for distribution approximation.

#### IV. EXPERIMENTAL DATASETS AND PARAMETER SETTINGS

In this section, we describe the experimental datasets in detail, including three benchmark datasets and the real-world gasoline blending process NIR. In addition, comparison methods and corresponding parameters have also been given.

##### A. Experimental Datasets

1) *Shootout Dataset:* The shootout benchmark dataset was first published at the International Diffuse Reflectance Conference (IDRC, 2002) [49]. The spectra of 655 pharmaceutical tablets were scanned from two spectrometers (Foss and Multitab Spectroscanners) and recorded from 600 to 1898 nm at 2-nm intervals.<sup>1</sup> The NIR between 600 and 1800 nm was used to construct the active pharmaceutical ingredient analysis model.

<sup>1</sup>[Online]. Shootout: <http://www.idrcchambersburg.org/shootout2002.html>.

---

##### Algorithm 2 Proposed AR-SCN Algorithm

---

**Input:** Measured dataset  $\{X, y\}$ , Gibbs sampling parameters:  $N_s$  and  $bi$ , initial block size  $\Delta_0$ ;

**Output:** Hidden layer output weight  $\beta$ ;

- 1: Initialize maximum number of hidden nodes  $P$ , maximum iteration times  $K$ , tolerance error  $\delta$ ;
- 2: Construct SCN based on the **Algorithm 1** and obtain the hidden output  $H$ ;
- 3: **While**  $i \leq N_s$ , **do**
- 4: Generate sample of  $\beta$ :  $\beta \leftarrow \mathcal{N}(D^{-1}H^T y, \sigma^2 D^{-1})$ ;
- 5: Generate sample of  $\sigma^2$ :
- 6:  $\sigma^2 \leftarrow \mathcal{IG}(N/2, y^T(I_N - H\Lambda_*^{-1}H^T)y/2)$ ;
- 7: Update  $\beta$ ,  $\sigma^2$ ,  $v$ ;
- 8: Generate sample of  $\Lambda$ :
- 9:  $\lambda_j^2 \leftarrow \mathcal{IG}(1, 1/v_j + \beta_j^2/2\tau^2\sigma^2)$ ;
- 10:  $\Lambda \leftarrow diag(\lambda_1^2, \lambda_2^2, \dots, \lambda_L^2)$ ;
- 11: Generate sample of  $v$ :  $v_j \leftarrow \mathcal{IG}(1, 1 + 1/\lambda_j^2)$ ;
- 12: Update  $\tau^2$  and generate sampl:  $\xi \leftarrow \mathcal{IG}(1, 1 + 1/\tau^2)$ ;
- 13: **If**  $i > bi$ , **do**
- 14:     Store the learning estimate of  $\beta$ ;
- 15: **End If**
- 16: **End While**
- 17: Output  $\beta$  of AR-SCN.

---

2) *Diesel Fuel Dataset:* The diesel fuel dataset was measured by the Southwest Research Institute. It consisted of 784 samples and six properties: bp50 (the boiling point at 50% recovery), cetane number (CN), d4052 (density), freeze (freezing temperature of the fuel), total (total aromatics), and visc (viscosity). The NIRs were collected from 750 to 1550 nm with 2-nm intervals.<sup>2</sup> It is worth noting that not all properties were measured for all samples, only the CN with 381 samples is used in this article.

3) *Public Gasoline Dataset:* The public gasoline dataset is often used as benchmark data for NIR analysis, which contains reflectance spectra of 60 samples with wavelengths from 900 to 1700 nm. In addition, the octane number was measured as the chemical property in this dataset [50].

4) *Gasoline Blending Dataset:* The gasoline blending dataset was collected from the NIR-based rapid evaluation system of the gasoline blending process in a refinery in China. The schematic of the gasoline blending and optimization system is illustrated in Fig. 4(a). The whole system comprises component tanks, additive agent system, optimizing controller system, blend header, product tanks, NIR analysis system, and so on. First, several different blending components are injected into the mixing system in preset proportion through the flow control system. Then, the passivating agent is added by the additive agent system, and blending begins. Finally, the blended gasoline is injected into the product tanks, and the physicochemical properties of the gasoline are evaluated before entering the market. During the actual gasoline blending process, the research octane number (RON) of gasoline must be measured in real time to ensure product quality and reduce production costs. During the production process, the online

<sup>2</sup>[Online]. Diesel fuel: <http://www.eigenvector.com/data/SWRI/index.html>.

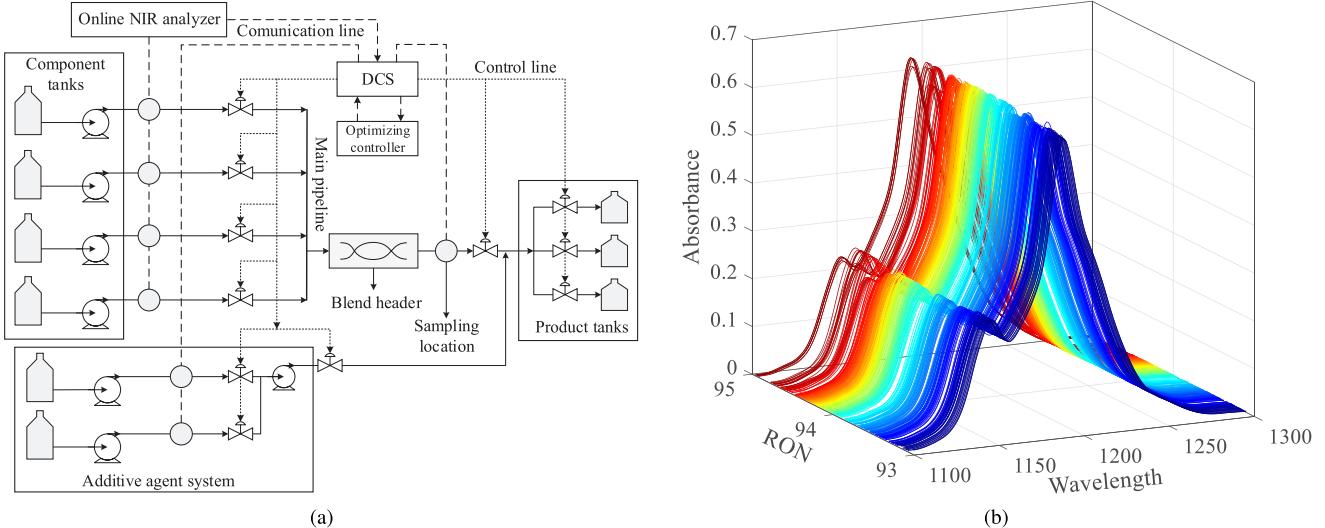


Fig. 4. Real-world gasoline blending process. (a) Diagram of the NIR-based fast evaluation system. (b) Real measured NIR spectra.

NIR analyzer scans the blended gasoline spectra and quickly predicts RON through the pre-established analysis model. As shown in Fig. 4(a), the guidewave grating NIR spectrometer (Guidewave M412) was used to collect spectra of online gasoline samples in the 1000–1600-nm wavelength range. To reduce the interference of noise and irrelevant variables, as shown in Fig. 4(b), the spectra between 1100 and 1300 nm were used for modeling analysis in this work.

Before building the NIR prediction model, all measurement datasets were randomly divided into a training set (70%) and a testing set (30%). In addition, the sample indices of different methods are kept consistent within a trial for a fair comparison. Furthermore, to verify the robustness of other methods, potential outliers (Nos. 11, 19, 122, 126, 127, 145, 267, 295, 294, 342, 313, 341, and 343 in shoot-out and Nos. 2, 9, 13, 20, 23, 37, 108, 122, 150, 151, 152, 235, 246, and 250 in diesel fuel dataset) were not removed in this article [1].

### B. Comparative Methods

Under the existing randomized learning framework, the following representative methods are selected for comparison.

1) *RVFL*: It is a classic randomized multilayer perceptron with a single hidden layer, where the input parameters are chosen randomly from a fixed interval, and the output parameters are usually obtained using a closed-form solution [18]. In addition, the optimal hidden layer width is generally obtained by cross validation.

2) *SCN*: It is a single hidden layer feedforward network in which the model structure is determined under the inequality supervisory mechanism [22]. In this work, the output weights are calculated using the global least-squares method.

3) *BSCN*: It is a new SCN framework for data analysis that allows learners to add fixed-size batches of nodes during the construction process [26]. The output weight solution of BSCN is the same as that of the SCN method.

4) *RSCN*: Robust SCN (RSCN) is a novel SCN method under the SBL framework, in which the prior distribution of the output parameters adopts the Gaussian distribution, and

the output of the hidden layer is determined by the Bayesian ridge regression algorithm [39].

5) *BSSCN*: Bayesian-based sparse SCN (BSSCN) is a hierarchical-Bayesian-based SCN for solving model complexity and ill-posed problems of high-dimensional datasets. Unlike the RSCN method, the Laplacian distribution is adopted as the prior distribution of the output weights [36].

6) *GM-SCN*: Generalized M-estimation SCN (GM-SCN) is a novel SCN, in which the GM method was adopted to reduce the influence of leverage points, and the  $L_2$  regularization was employed for robust output weight learning [29].

### C. Parameter Setting

1) *Setting of Tolerance Error  $\delta$* : To test the effect of tolerance error  $\delta$  on the model performance, we compare the performance of different  $\delta$  on the real-world NIR dataset to determine the expected tolerance error  $\delta$ . Fig. 5(a) shows the average of the coefficient of determination ( $R^2$ ) on the testing dataset over 50 independent trials. It can be seen that the  $R^2$  increases rapidly when  $\delta \in (0, 0.17]$  and then begins to decrease slowly as  $\delta$  increases. Accordingly, the tolerance error  $\delta$  is set to 0.17 for the measured gasoline blending NIR. In addition, the performance with different  $\delta$ 's is also compared for the benchmark datasets, and the selected parameters of SCN-based methods are given in Table I, where superscripts *P* and *M* represent the public gasoline and the measured gasoline blending datasets, respectively.

2) *Setting of Initial Block Size  $\Delta_0$* : Fig. 5(b) shows the effect of the initial block size  $\Delta_0$  of the AR-SCN method on the learning performance of the measured gasoline NIR. It can be seen that an overly large  $\Delta_0$  easily produces an unstable learner, while an overly small  $\Delta_0$  increases computation time. Thus,  $\Delta_0$  is set to 3 for the real-world dataset. In addition, we compare the effect of  $\Delta_0$  for other datasets and summarized in Table I.

3) *Setting of Sampling Times  $N_s$* : The performance difference of different sampling times  $N_s$  is shown in Fig. 5(c). It can be seen that small  $N_s$  easily obtains an unstable model,

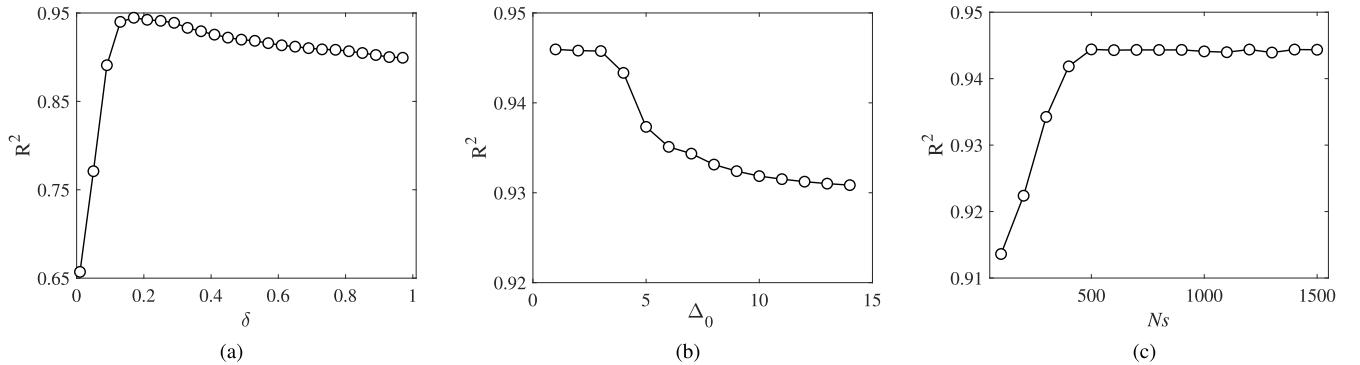


Fig. 5. Empirical parameter settings of the proposed AR-SCN for the real-world dataset. (a) Tolerance error  $\delta$ . (b) Initial block size  $\Delta_0$ . (c) Sampling times  $N_s$ .

TABLE I  
MODEL PARAMETER SETTING OF DIFFERENT DATASETS

Methods	Parameters	Shootout	Diesel fuel	Gasoline <sup>P</sup>	Gasoline <sup>M</sup>
RVFL	$L$	42	40	11	17
SCN-based	$\delta$	4.10	5.00	0.18	0.17
BSCN	$\Delta_B$	2	3	4	3
GM-SCN	$\alpha$	0.51	0.73	0.46	0.18
AR-SCN	$\Delta_0, N_s$	4, 600	3, 800	2, 400	3, 500

and overly large  $N_s$  will increase training time. Therefore,  $N_s$  of the measured dataset is set to 500. In addition, we also determine the proper  $N_s$  for other datasets and summarize them in Table I. The hidden layers width ( $L$ ) of the RVFL, the incremental block size ( $\Delta_B$ ) of the BSCN, and the hyperparameter ( $\alpha$ ) of the GM-SCN are determined by cross validation in different datasets. The termination condition of the RSCN and the BSSCN is set as  $10^{-6}$  as suggested in [36] and [39]. In addition,  $b_i$  of the AR-SCN is set as 100 in all datasets. Table I summarizes the model parameters of different methods of experimental datasets.

## V. RESULTS

In this section, three benchmark datasets and one real-world dataset are used to verify the performance of the proposed algorithm. The root-mean-squared error (RMSE) and  $R^2$  are used for model performance comparison. The experiments were repeated 100 times, and the statistical results were summarized. All experiments were written in MATLAB (version 2021b, MathWorks, Inc.) and conducted on a computer with a Microsoft Windows 10 operating system.

### A. Case Studies on Benchmark Datasets

To compare the differences between different incremental learning strategies in the network construction process, the variation trends of prediction error with hidden layer width and iteration times of different incremental learning methods are shown in Fig. 6, where SCN, BSCN, and AR-SCN denote the point incremental, block incremental, and adaptive incremental strategies, respectively. Due to the increase of network nodes in batches, the BSCN can quickly reduce the

prediction error with the least number of iterations. However, compared with the SCN and AR-SCN, the network width of the BSCN increases by 40.00%, 15.00%, and 24.40% in benchmark datasets. Although the SCN converges slowly throughout the learning process, it finally obtains networks with the smallest hidden layer width. The proposed AR-SCN method converges as fast as the BSCN in the initial stage. Finally, the same hidden layer width as the SCN is obtained, indicating that the proposed adaptive incremental learning strategy can effectively accelerate the construction process while obtaining a moderate-sized model.

To further compare the performance of the different methods, Table II shows the mean and standard deviation of  $R^2$  and RMSE of the prediction set ( $R_p^2$  and RMSEP) for 100 independent trials. As summarized in Table II, the proposed AR-SCN method outperforms other methods on three benchmark datasets. Specifically, compared with the SCN, the  $R_p^2$  of the AR-SCN improves by 0.97%, 3.39%, and 7.12%. The performance of the RVFL based on unsupervised incremental learning is lower than that of the SCN and BSCN methods. For the BSCN, the redundant structure caused by block increment makes the prediction performance slightly lower than that of the SCN. It is worth noting that apart from performing similar to the SCN and BSCN methods on the diesel fuel dataset, the RSCN and BSSCN methods did not achieve significant improvements on the public gasoline and shootout datasets, which indicates that the Gaussian and Laplacian prior distributions have limited effect on high-dimensional small-sample spectra and are sensitive to outliers. Compared with the RSCN, the performance improvement of the GM-SCN method on the public gasoline and shootout datasets is mainly attributed to the leverage sample estimation. However, sample weighting based on independent samples performs poorly in the diesel fuel dataset with large fluctuation ranges. In addition, the best results with a relatively small standard deviation achieved by the AR-SCN method indicate that it is more robust than other methods when dealing with outliers.

To evaluate the computational efficiency of the proposed AR-SCN method, the mean and standard deviation of training time are summarized in Table III. It can be seen that the training time of the BSCN is less than other methods by

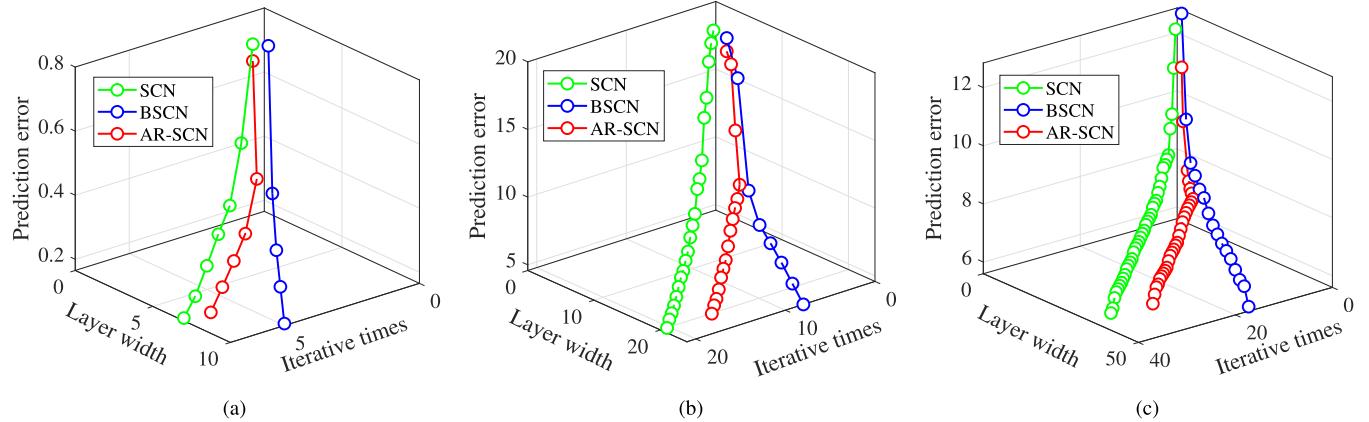


Fig. 6. Prediction error with the hidden layer width and iterations of different incremental learning methods. (a) Public gasoline. (b) Diesel fuel. (c) Shootout datasets.

TABLE II  
PREDICTION RESULTS OF COMPARATIVE METHODS ON BENCHMARK DATASETS

Method	Public gasoline		Diesel fuel		Shootout	
	R <sub>p</sub> <sup>2</sup>	RMSEP	R <sub>p</sub> <sup>2</sup>	RMSEP	R <sub>p</sub> <sup>2</sup>	RMSEP
RVFL	0.9447 ± 0.0586	0.3162 ± 0.0748	0.8621 ± 0.1298	7.9799 ± 4.6041	0.7398 ± 0.2071	9.0063 ± 3.9507
SCN	0.9585 ± 0.0178	0.2795 ± 0.0575	0.8831 ± 0.1002	6.6227 ± 2.1227	0.8635 ± 0.0982	6.1315 ± 2.1959
BSCN	0.9481 ± 0.0529	0.2991 ± 0.1137	0.8775 ± 0.1159	6.9154 ± 2.2628	0.8614 ± 0.0871	6.0913 ± 2.5484
RSCN	0.9214 ± 0.0554	0.3936 ± 0.1221	0.8848 ± 0.0730	6.7246 ± 1.6739	0.7457 ± 0.2598	7.8344 ± 3.7666
BSSCN	0.9118 ± 0.0681	0.4116 ± 0.1403	0.8857 ± 0.0964	6.7269 ± 1.5977	0.7406 ± 0.2478	7.8561 ± 3.7527
GM-SCN	0.9615 ± 0.0184	0.2648 ± 0.0676	0.8614 ± 0.0694	8.3540 ± 2.0636	0.8851 ± 0.1439	5.2706 ± 2.8856
AR-SCN	<b>0.9678 ± 0.0179</b>	<b>0.2491 ± 0.0599</b>	<b>0.9130 ± 0.0567</b>	<b>5.8679 ± 1.4593</b>	<b>0.9250 ± 0.1076</b>	<b>4.3073 ± 1.9712</b>

TABLE III

COMPUTATIONAL TIME CONSUMPTION OF DIFFERENT METHODS ON BENCHMARK DATASETS

	Public gasoline	Diesel fuel	Shootout
RVFL	0.4361 ± 0.0067	2.9075 ± 0.0435	16.3614 ± 0.5487
SCN	0.1370 ± 0.0178	1.0297 ± 0.4162	11.9644 ± 1.2406
BSCN	<b>0.0842 ± 0.0161</b>	<b>0.5195 ± 0.1951</b>	<b>4.2886 ± 1.2609</b>
RSCN	0.1720 ± 0.0176	5.0055 ± 0.6689	37.0931 ± 3.6845
BSSCN	0.2427 ± 0.0205	4.9895 ± 0.6998	33.1648 ± 4.1919
GM-SCN	0.1684 ± 0.0174	1.3542 ± 0.5297	6.8111 ± 0.9962
AR-SCN	0.3192 ± 0.0364	6.4998 ± 0.9936	43.0601 ± 4.2309

adding block nodes. The RVFL is slightly longer than the SCN and the BSCN due to the need for cross validation to determine the appropriate number of nodes. In addition, compared with the SCN, the training time of the RSCN and BSSCN methods dramatically increases due to the iterative process of model parameters inference. Compared with the RSCN and BSSCN methods, the GM-SCN method takes less computational time on benchmark datasets, mainly because it only includes distance calculation related to the number of samples. Due to the MCMC sampling process, the parameter optimization process of the AR-SCN method is slightly longer than other methods. However, it performs best regarding key properties prediction on all benchmark datasets.

### B. Case Studies on Real-World Dataset

The prediction error, iterative times, and hidden layer width of the gasoline blending dataset are shown in Fig. 7 to

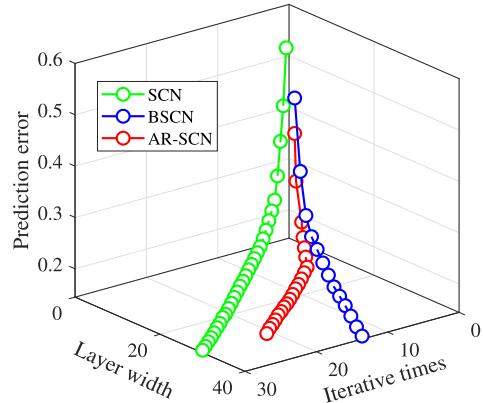


Fig. 7. Prediction error with the hidden layer width and iterations of different incremental learning methods.

illustrate the difference between different learning strategies. It can be observed that there are similar convergence cures between the SCN and AR-SCN, and the same minimum prediction error with the smallest layer width is obtained. On the contrary, the BSCN quickly reduces the prediction error by adding nodes in batches. Combined with the calculation time of 100 random experiments summarized in Table IV, it is shown that compared with the SCN, the proposed adaptive incremental learning method can reduce the construction time by 26.76% and has similar testing times. Due to the limitations of random incremental learning under a fixed random parameter range, the RVFL method requires

TABLE IV

COMPUTATIONAL EFFICIENCY, HIDDEN LAYER WIDTH, AND PREDICTION PERFORMANCE OF DIFFERENT METHODS ON THE REAL-WORLD DATASET

	Construction time (s)	Optimization time (s)	Testing time (s)	Layer width	$R_T^2$	RMSET	$R_P^2$	RMSEP
RVFL	0.2676 $\pm$ 0.0167	\	0.0010 $\pm$ 0.0004	47.79 $\pm$ 1.48	0.9345	0.1944	0.9165	0.2251
SCN	0.1618 $\pm$ 0.0248	\	0.0012 $\pm$ 0.0007	26.43 $\pm$ 2.95	0.9419	0.1868	0.9236	0.2141
BSCN	<b>0.0691 <math>\pm</math> 0.0115</b>	\	<b>0.0009 <math>\pm</math> 0.0004</b>	32.85 $\pm$ 3.20	0.9443	0.1837	0.9186	0.2185
RSCN	0.1626 $\pm$ 0.0261	0.4532 $\pm$ 0.0232	0.0012 $\pm$ 0.0006	26.52 $\pm$ 2.82	0.8078	0.3462	0.8018	0.3473
BSSCN	0.1659 $\pm$ 0.0261	0.4507 $\pm$ 0.0587	0.0011 $\pm$ 0.0003	26.68 $\pm$ 2.91	0.8121	0.3402	0.7923	0.3514
GM-SCN	0.1763 $\pm$ 0.0285	<b>0.0010 <math>\pm</math> 0.0005</b>	0.0010 $\pm$ 0.0002	<b>26.25 <math>\pm</math> 2.66</b>	0.8207	0.3394	0.7196	0.3985
AR-SCN	0.1185 $\pm$ 0.0258	0.4702 $\pm$ 0.0254	0.0013 $\pm$ 0.0003	27.09 $\pm$ 2.64	<b>0.9617</b>	<b>0.1487</b>	<b>0.9506</b>	<b>0.1497</b>

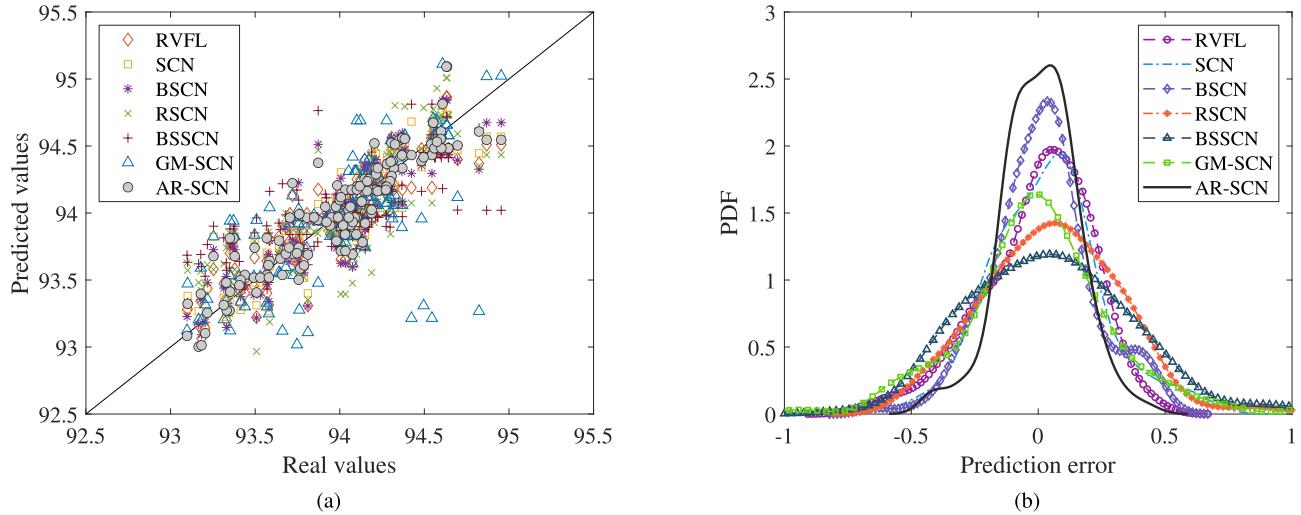
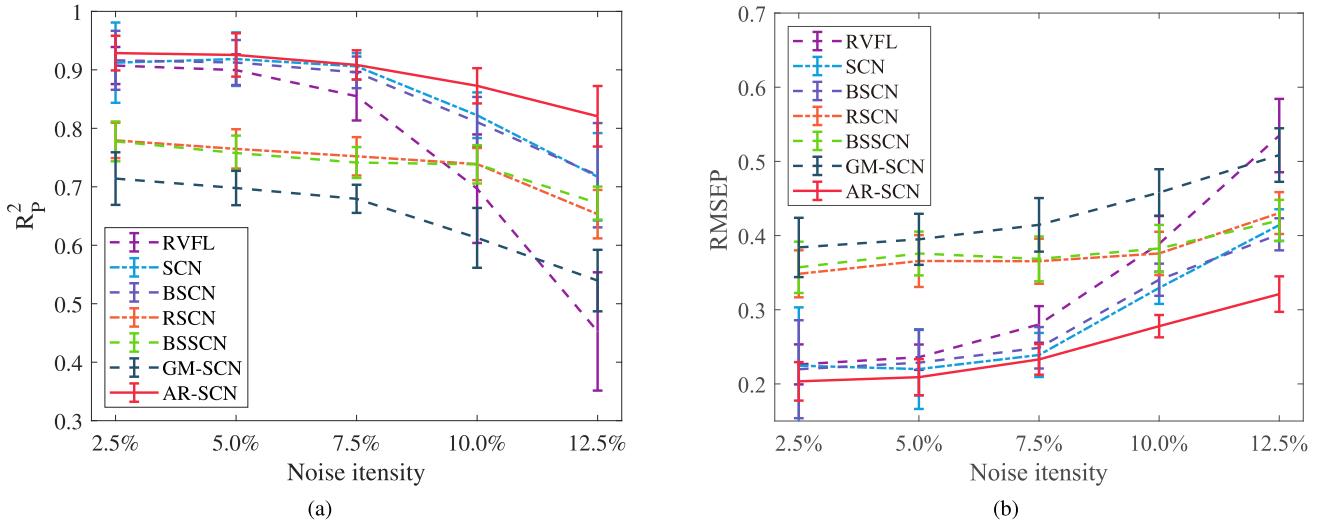


Fig. 8. Prediction results of real-world spectra. (a) Scatter plots of RON prediction. (b) PDF of prediction residuals of comparative methods.

Fig. 9. Model performance comparison under different noise intensities. (a)  $R_P^2$ . (b) RMSEP.

more time to build a complex model to predict the RON of the gasoline blending process. In contrast, the BSCN based on batch incremental learning significantly reduces model construction time but inevitably causes model redundancy. Due to the time consumption of sample weighting, the model construction time of the GM-SCN method is slightly higher than that of other SCN-based methods, but its direct parameter solution process takes less time. Compared to the RSCN and

BSCN, experimental results demonstrate that the proposed AR-SCN approach does not increase model complexity while accelerating the construction.

To further verify the effectiveness of the AR-SCN method in the actual gasoline blending process, the scatterplot of the RON prediction results and the probability density function (pdf) of the prediction error shown in Fig. 8 are used to demonstrate the performance difference between different methods

in a single experiment. As shown in Fig. 8(a), although other methods can accurately predict RON, the proposed AR-SCN method still outperforms other methods. Meanwhile, from the distribution difference of prediction residuals shown in Fig. 8(b), it can be known that the AR-SCN has a larger pdf near the origin than other methods, which means that the proposed AR-SCN has a higher probability of obtaining a better prediction model. In contrast, there are flatter distribution errors between other methods, which means that these methods are prone to more significant prediction errors in the actual process. Meanwhile, based on the statistical results of model performance shown in Table IV, it can be found that the  $R_p^2$  of the AR-SCN method is greater than that of comparative methods, while the RMSE is smaller than the existing methods. Similarly, the RSCN and BSSCN are inferior to the SCN and BSCN, which is consistent with the discussion in Section III: uniform coefficient shrinkage affects the robustness and generalization of NIR model. It is worth noting that the GM-SCN method performs poorly on the gasoline blending data, which may be attributed to the response distribution shift caused by the response weighting only based on the sample spectra distance. Furthermore, the nonsparse shrinkage and biased estimation of  $L_2$  regularization is also difficult to robustly estimate the output weights of the constructed SCN.

The robustness of the multivariate analysis method is of great significance for NIR application. To verify the robustness of different methods,  $\zeta\%$  ( $\zeta = 2.5, 5, 7.5, 10, 12.5$ ) of mean spectra intensity is introduced to the real NIR dataset, which is produced as  $\zeta\% \times \bar{x} \times \text{randn}(N, M)$  and  $\text{randn}(N, M)$  denotes a pseudorandom number generator for generating real dataset-sized random numbers. The statistical results of the  $R_p^2$  and RMSEP with different noise intensities of comparative methods are illustrated in Fig. 9. It can be seen that the methods based on Bayesian estimation are more stable than the methods based on a linear solution. Although the RSCN and BSSCN methods are relatively stable under the first four noise disturbances, their prediction accuracy is still inferior to the proposed AR-SCN method. The GM-SCN method performs poorly under different noise intensities, which validates that response weighting based on spectral distance may introduce interference and degrade model robustness.

The above comparison results on real NIR dataset show that the proposed AR-SCN method performs well in RON prediction with higher accuracy than existing SCN methods. In addition, the proposed method shows better robustness in different noise intensities. The experimental results show that the proposed method can provide reliable and accurate information for predicting key physicochemical properties in the gasoline blending process.

## VI. CONCLUSION

This article presents a novel method for NIR multivariate analysis and verifies its effectiveness. The adaptive incremental learning method based on the prediction residual is developed to adjust incremental block size during the learning process, which can accelerate the model construction process and avoid

complex structures. The robust estimation method based on the global-local shrinkage is proposed for output weight learning, promoting global sparsity while avoiding overshrinkage for the local significant weights. Experimental results on three benchmark datasets demonstrate that the proposed AR-SCN method can quickly build compact models with better predictive performance. Experimental results on the real-world NIR dataset also demonstrate that the proposed AR-SCN method has a better performance and robustness in the rapid prediction of key attributes of gasoline blending process, which can provide useful information for real-time control and decision-making of the refining process.

It is worth noting that the proposed method still has some limitations, such as the initial block size being determined by trial and error, which may bring subjective interference into NIR modeling. In addition, the iterative sampling estimation of output weights is relatively complex, which limits the application of this method in large-scale NIR datasets. To alleviate these problems, future works will focus on developing problem-dependent initialization strategies to reduce human intervention and investigating advanced inference methods for complex network parameter estimation in NIR analysis.

## REFERENCES

- [1] K. Wang, W. Du, and J. Long, "Near-infrared wavelength-selection method based on joint mutual information and weighted bootstrap sampling," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5884–5894, Sep. 2020.
- [2] K. He, M. Zhong, J. Fang, and Y. Li, "Biased minimax probability machine-based adaptive regression for online analysis of gasoline property," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2799–2808, Apr. 2020.
- [3] R. G. Afkhami, F. R. Walker, S. Ramadan, and S. Johnson, "A dynamic model of brain hemodynamics in near-infrared spectroscopy," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 7, pp. 2103–2109, Jul. 2020.
- [4] H. Chen, H. Qiao, L. Xu, Q. Feng, and K. Cai, "A fuzzy optimization strategy for the implementation of RBF LSSVR model in Vis-NIR analysis of pomelo maturity," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5971–5979, Nov. 2019.
- [5] H. Li, P. Wu, J. Dai, and X. Zou, "A Monte Carlo resampling based multiple feature-spaces ensemble (MFE) strategy for consistency-enhanced spectral variable selection," *Analytica Chim. Acta*, vol. 1279, Oct. 2023, Art. no. 341782.
- [6] H. Yu, W. Du, Z.-Q. Lang, K. Wang, and J. Long, "A novel integrated approach to characterization of petroleum naphtha properties from near-infrared spectroscopy," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
- [7] L. Zou, X. Yu, M. Li, M. Lei, and H. Yu, "Nondestructive identification of coal and gangue via near-infrared spectroscopy based on improved broad learning," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8043–8052, Oct. 2020.
- [8] Y. Wu, Y. Liu, X. Li, Z. Huang, and D. Han, "Gasoline octane number prediction from near-infrared spectroscopy with an ANN-based model," *Fuel*, vol. 318, Jun. 2022, Art. no. 123543.
- [9] Y. Yu, M. Yao, J. Huang, and X. Xiao, "When process analysis technology meets transfer learning: A model transfer strategy between different spectrometers for quantitative analysis," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, Art. no. 2507619.
- [10] K. Cai, H. Chen, W. Ai, X. Miao, Q. Lin, and Q. Feng, "Feedback convolutional network for intelligent data fusion based on near-infrared collaborative IoT technology," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1200–1209, Feb. 2022.
- [11] J. Dong, B. Hou, L. Feng, H. Tang, K. C. Tan, and Y.-S. Ong, "A cell-based fast memetic algorithm for automated convolutional neural architecture design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9040–9053, Nov. 2023.

- [12] D. Passos and P. Mishra, "Perspectives on deep learning for near-infrared spectral data modelling," *NIR News*, vol. 33, nos. 7–8, pp. 9–12, Nov. 2022.
- [13] T.-Y. Kwok and D.-Y. Yeung, "Objective functions for training new hidden units in constructive neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1131–1148, Sep. 1997.
- [14] X. Wu, P. Rózycki, and B. M. Wilamowski, "A hybrid constructive algorithm for single-layer feedforward networks learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1659–1668, Aug. 2015.
- [15] M. Li and D. Wang, "Insights into randomized algorithms for neural networks: Practical issues and common pitfalls," *Inf. Sci.*, vol. 282, pp. 170–178, Mar. 2017.
- [16] S. Scardapane and D. Wang, "Randomness in neural networks: An overview," *WIREs Data Mining Knowl. Discovery*, vol. 7, no. 2, pp. 1–18, Mar. 2017.
- [17] L. Zhang and P. N. Suganthan, "A survey of randomized algorithms for training neural networks," *Inf. Sci.*, vols. 364–365, pp. 146–155, Oct. 2016.
- [18] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: Theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, May 1992.
- [19] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, Apr. 1994.
- [20] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.
- [21] M. Li and D. Wang, "2-D stochastic configuration networks for image data analytics," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 359–372, Jan. 2021.
- [22] D. Wang and M. Li, "Stochastic configuration networks: Fundamentals and algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3466–3479, Oct. 2017.
- [23] D. Wang and M. Li, "Deep stochastic configuration networks with universal approximation property," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.
- [24] J. Sun and C.-S. Leung, "Regularization effect of random node fault/noise on gradient descent learning algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2619–2632, May 2023.
- [25] K. Li, L. Zhang, and J. Qiao, "Multi-task stochastic configuration network with autonomous linking and its application in wastewater treatment processes," *Inf. Sci.*, vol. 662, Mar. 2024, Art. no. 120195.
- [26] W. Dai, D. Li, P. Zhou, and T. Chai, "Stochastic configuration networks with block increments for data modeling in process industries," *Inf. Sci.*, vol. 484, pp. 367–386, May 2019.
- [27] W. Dai, X. Zhou, D. Li, S. Zhu, and X. Wang, "Hybrid parallel stochastic configuration networks for industrial data analytics," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2331–2341, Apr. 2022.
- [28] K. Li, J. Qiao, and D. Wang, "Online self-learning stochastic configuration networks for nonstationary data stream analysis," *IEEE Trans. Ind. Informat.*, vol. 20, no. 3, pp. 3222–3231, Mar. 2024.
- [29] K. Sun, L. Zhao, P. Tian, J. Zhao, and D. Wang, "Prediction of X-ray fluorescence copper grade using regularized stochastic configuration networks," *Inf. Sci.*, vol. 659, Feb. 2024, Art. no. 120098.
- [30] C. Cui and D. Wang, "High dimensional data regression using lasso model and neural networks with random weights," *Inf. Sci.*, vol. 372, pp. 505–517, Dec. 2016.
- [31] H. Ye, F. Cao, and D. Wang, "A hybrid regularization approach for random vector functional-link networks," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112912.
- [32] E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted  $\ell_1$  minimization," *J. Fourier Anal. Appl.*, vol. 44, pp. 877–905, Oct. 2008.
- [33] F. Cao, Y. Tan, and M. Cai, "Sparse algorithms of random weight networks and applications," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2457–2462, Apr. 2014.
- [34] C. De Mol, E. De Vito, and L. Rosasco, "Elastic-net regularization in learning theory," *J. Complex.*, vol. 25, no. 2, pp. 201–230, Apr. 2009.
- [35] S. Scardapane, D. Wang, and A. Uncini, "Bayesian random vector functional-link networks for robust data modeling," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2049–2059, Jul. 2018.
- [36] J. Lu, J. Ding, C. Liu, and T. Chai, "Hierarchical-Bayesian-based sparse stochastic configuration networks for construction of prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3560–3571, Aug. 2022.
- [37] I. Selesnick, "Sparse regularization via convex analysis," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4481–4494, Sep. 2017.
- [38] T. Park and G. Casella, "The Bayesian lasso," *J. Am. Stat. Assoc.*, vol. 103, no. 482, pp. 681–686, Jun. 2008.
- [39] J. Lu, J. Ding, X. Dai, and T. Chai, "Ensemble stochastic configuration networks for estimating prediction intervals: A simultaneous robust training algorithm and its application," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5426–5440, Dec. 2020.
- [40] A. Yan, J. Guo, and D. Wang, "Robust stochastic configuration networks for industrial data modelling with student's-t mixture distribution," *Inf. Sci.*, vol. 607, pp. 493–505, Aug. 2022.
- [41] N. Agarwala, J. Park, and A. Roy, "Horseshoe and Strawderman-Berger estimators for non-negative normal means," *Stat. Appl.*, vol. 18, pp. 317–332, Sep. 2020.
- [42] C. Carvalho, N. Polson, and J. Scott, "Handling sparsity via the horseshoe," in *Proc. Int. Conf. Artif. Intell. Stat.*, FL, USA, 2009, pp. 73–80.
- [43] E. Makalic and D. F. Schmidt, "A simple sampler for the horseshoe estimator," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 179–182, Jan. 2016.
- [44] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.
- [45] A. Bhadra, J. Datta, Y. Li, N. G. Polson, and B. T. Willard, "Prediction risk for the horseshoe regression," *J. Mach. Learn. Res.*, vol. 20, no. 78, pp. 1–39, Mar. 2019.
- [46] Q. Song and F. Liang, "Nearly optimal Bayesian shrinkage for high-dimensional regression," *Sci. China Math.*, vol. 66, no. 2, pp. 409–442, Oct. 2022.
- [47] A. Bhadra, J. Datta, N. G. Polson, and B. Willard, "Lasso meets horseshoe: A survey," *Stat. Sci.*, vol. 34, no. 3, pp. 405–427, Aug. 2019.
- [48] B. Rajaratnam, D. Sparks, K. Khare, and L. Zhang, "Uncertainty quantification for modern high-dimensional regression via scalable Bayesian methods," *J. Comput. Graph. Statist.*, vol. 28, no. 1, pp. 174–184, Sep. 2018.
- [49] D. W. Hopkins, "Shoot-out 2002: Transfer of calibration for content of active in a pharmaceutical tablet," *NIR News*, vol. 14, no. 5, pp. 10–13, Oct. 2003.
- [50] J. H. Kalivas, "Two data sets of near infrared spectra," *Chemometric Intell. Lab. Syst.*, vol. 37, no. 2, pp. 255–259, Jun. 1997.



**Yuqiang Li** received the M.S. degree in control theory and control engineering from Jiangsu University, Zhenjiang, China, in 2021. He is currently pursuing the Ph.D. degree in control science and engineering with East China University of Science and Technology, Shanghai, China.

His current research interests include chemometrics analysis, statistical learning with sparsity, and industrial process modeling.



**Wenli Du** (Senior Member, IEEE) received the B.S. and M.S. degrees in chemical process control from Dalian University of Technology, Dalian, China, in 1997 and 2000, respectively, and the Ph.D. degree in control theory and control engineering from East China University of Science and Technology, Shanghai, China, in 2005.

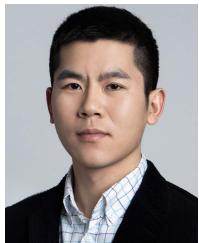
She is currently a Professor and the Dean of the Graduate School and the Vice Dean of the Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology. Her current research interests include control theory and applications, system modeling, advanced control, and process optimization.

Dr. Du was a recipient of the National Science Fund for Distinguished Young Scholars, five State Science and Technology Progress Awards, and 12 first prizes of Provincial and Ministerial-Level Science and Technology Awards. She has served as the Executive Director for Chinese Association of Automation and Chinese Association of Artificial Intelligence. She is an Associate Editor of seven international journals, including *Industrial and Engineering Chemistry Research*, *Computers and Chemical Engineering*, *Complex and Intelligent Systems*, and *Frontiers in Chemical Engineering*.



**Minglei Yang** received the B.S. and Ph.D. degrees in chemical engineering from the School of Chemical Engineering, East China University of Science and Technology (ECUST), Shanghai, China, in 2007 and 2012, respectively.

He is currently an Associate Professor with the School of Information Science and Technology, ECUST. His research interests include industrial process modeling, real-time optimization, large-scale optimization of decision-making, and rapid characterization of crude oil through near infrared (NIR).



**Xinjie Wang** received the Ph.D. degree from the College of Information Sciences and Technology, Donghua University, Shanghai, China, in 2020.

From July 2021 to October 2023, he held a post-doctoral position at the Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai. He is currently with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China. His current research interests include computational modeling of plasticity and data-driven evolutionary optimization.



**Yunmeng Zhao** received the B.S. degree in chemistry from Nanjing University, Nanjing, China, in 2014, and the Ph.D. degree in chemical engineering from Monash University, Melbourne, VIC, Australia, in 2019.

He is currently a Faculty Member with East China University of Science and Technology, Shanghai, China. His research interests encompass process analytical technology, process systems engineering, and soft electronics.