Electronic Supplementary Material

A hybrid modeling strategy based on deep learning surrogate models for accurate process multi-objective optimization of isooctanol oxidation

Xin Zhou[1], Zhibo Zhang[2], Mengzhen Zhu[2], Hui Zhao[2], Hao Yan (✉)[2], Chaohe Yang[2]

1 College of Chemistry and Chemical Engineering, Ocean University of China, Qingdao 266100, China

2 State Key Laboratory of Heavy Oil Processing, China University of Petroleum, Qingdao 266580, China

E-mail: haoyan@upc.edu.cn

Experiment

The data of octanol dehydrogenation to produce octanal are shown in Table S0

Table S0 -octanol dehydrogenation on the $Cu/La_2O_2CO_3$ catalyst at 373-413 K.[1,2]

Entry	T (K)	Conv. (%)	Select. (%)	Yield (%)
1	373	8	99	8
2	383	18	99	18
3	393	25	99	25
4	403	27	99	27
5	408	40	99	40
6	413	65	99	64

Reaction conditions: 1-octanol (2 mmol), mesitylene (8 ml), catalyst (200 mg), styrene (8 mmol), 12 h, N_2 atmosphere.

1. Reaction kinetic and process model

The dehydrogenation of octanol to octanal is a typical catalytic dehydrogenation reaction, usually carried out under appropriate catalysts and reaction conditions. To derive the reaction

kinetics equation, we need to consider factors such as reaction mechanism, reaction order, and catalyst surface reactions. Below is a possible derivation process:

1. Assumption of Reaction Mechanism:

Octanol (1-octanol) undergoes dehydrogenation on the catalyst surface to form octanal (1-octanal). Assume the reaction follows these steps:

$$C_8H_{17}OH + * \leftrightharpoons C_8H_{17}O^* + H^* \quad (1)$$

$$C_8H_{17}O^* \rightarrow C_8H_{16}O + * \quad (2)$$

$$2H^* \rightarrow H_2 + 2* \quad (3)$$

where * represents active sites on the catalyst.

2. Assuming Step 2 is the Rate-Determining Step:

If the second step (dehydrogenation of octanol to octanal) is the rate-determining step, then the reaction rate can be expressed as:

$$r = k \cdot \theta_{C_8H_{17}O}$$

where k is the rate constant and $\theta_{C_8H_{17}O}$ is the surface coverage of octanol on the catalyst.

3. Expression for Surface Coverage:

Using the Langmuir adsorption isotherm to represent surface coverage, and assuming adsorption equilibrium is achieved, we have:

$$\theta_{C_8H_{17}O} = \frac{K \cdot P_{C_8H_{17}OH}}{1 + K \cdot P_{C_8H_{17}OH}}$$

where K is the adsorption equilibrium constant and $P_{C_8H_{17}OH}$ is the partial pressure of octanol.

4. Combining the Rate Equation and Surface Coverage:

Substitute the expression for surface coverage into the rate equation:

$$r = k \cdot \frac{K \cdot P_{C_8H_{17}OH}}{1 + K \cdot P_{C_8H_{17}OH}}$$

5. Simplifying the Kinetic Equation:

To simplify the equation, we can introduce a new rate constant k' = kK, giving:

$$r = \frac{k' \cdot P_{C_8H_{17}OH}}{1 + K \cdot P_{C_8H_{17}OH}}$$

6. Form of the Kinetic Equation

The final reaction kinetics equation is:

$$r = \frac{k' \cdot P_{C_8H_{17}OH}}{1 + K \cdot P_{C_8H_{17}OH}}$$

This is the typical Langmuir-Hinshelwood kinetics model, suitable for describing catalytic surface reactions. This kinetics equation indicates that at low partial pressures, the reaction rate is proportional to the partial pressure of octanol; whereas at high partial pressures, the reaction rate approaches a constant value, showing saturation behavior. This equation can be used for further studies on the specific mechanism of the reaction, optimizing catalysts, and reaction conditions.

The capacity of the octanal to octanol is 10000 tons per year. The process simulation of octanol dehydrogenation to octanal involves modeling and simulating the chemical reaction

process to optimize reaction conditions, enhance yield, and improve efficiency. First, it is essential to collect input data, including the physical and chemical properties of the raw material (octanol) such as density, viscosity, boiling point, and heat capacity, as well as the characteristics of the required catalyst like activity, selectivity, and lifespan. Determining the reaction mechanism and deriving the kinetic equation is crucial, utilizing experimental or literature data to obtain kinetic parameters like rate constants and adsorption equilibrium constants. Selecting an appropriate reactor type and designing the reactor's geometric parameters are also key steps. Using process simulation software (Aspen Plus), the reaction model is established, incorporating thermodynamic models (e.g., Peng-Robinson) to predict phase behavior and heat/mass transfer phenomena accurately. The reactor module is defined within the software, and kinetic equations and parameters are inputted.

Optimization of reaction conditions is performed by adjusting variables such as temperature, pressure, and feed flow rate to maximize the conversion of octanol and selectivity of octanal. The distribution of products and energy consumption under different operating conditions is analyzed. The product separation and purification design involves unit operations like distillation, extraction, or adsorption to separate and purify octanal from the reaction mixture. Separation modules are added to the simulation, and parameters are adjusted to enhance separation efficiency. An economic analysis and evaluation are conducted to estimate costs, including raw materials, energy, equipment, and operation costs, assessing the economic feasibility of the process and determining the economic benefits under optimal operating conditions. Through these steps, the process simulation achieves the dehydrogenation of octanol to octanal, optimizing reaction conditions and equipment design to improve the overall process's economic efficiency and feasibility, aiding engineers in making informed decisions during the design and operation phases.

Subsequently, Equation (1)-Equation (11) provide constraint conditions for the decision variables involved above. It is worth mentioning that, in order to ensure qualified products, equation (12) stipulates that the purity of isooctanal obtained after separation must be greater than 99.5%. C_{oct} represents the quality and purity of isooctanal. The boundaries are set based on experimental data and practical considerations, considering factors like feasibility, system stability, and goal achievement. For instance, the reaction condition Xr was selected by

referencing the iso-octanol process temperature and verifying it with preliminary experiments. The reaction pressure was set according to the experimental conditions, ensuring all key factors were covered.

$$s.t.: \quad g(X_r, X_s, Y_s, X_e, Y_n, X_i, X_o)=0 \tag{1}$$

$$G(X_r, X_s, Y_s, X_e, Y_n, X_i, X_o) \neq 0 \tag{2}$$

$$\underline{X}_r \leq X_r \leq \bar{X}_r \tag{3}$$

$$\underline{X}_s \leq X_s \leq \bar{X}_s \tag{4}$$

$$\underline{X}_e \leq X_e \leq \bar{X}_e \tag{5}$$

$$\underline{X}_i \leq X_i \leq \bar{X}_i \tag{6}$$

$$\underline{X}_o \leq X_o \leq \bar{X}_o \tag{7}$$

$$\underline{Y}_n \leq Y_n \leq \bar{Y}_n \tag{8}$$

$$\underline{Y}_s \leq Y_s \leq \bar{Y}_s \tag{9}$$

$$X_r, X_s, Y_s, X_e, X_i, X_o \in R^n \tag{10}$$

$$Y_n, Y_s \in N+ \tag{11}$$

$$C_{oct} \geq 99.5\% \tag{12}$$

The process simulation of octanol dehydrogenation to octanal involves several key steps to optimize reaction conditions, enhance yield, and improve efficiency. Initially, octanol is pumped into a preheater, where it is heated to approximately 100°C. The heated octanol then enters a fixed-bed reactor for the dehydrogenation reaction. Following the reaction, the gas-liquid mixture proceeds to a two-stage cooling and flash tank equipped with an external coil for effective cooling, facilitating the maximum separation of hydrogen gas. The liquid

phase is subsequently pumped into a distillation column, where any remaining gases and minor light impurities are separated. The distillation column's top section produces the desired octanal product, while unreacted octanol is collected at the bottom for recycling.

Through these steps, the simulation process aims to optimize reaction conditions and equipment design, thereby improving the overall economic efficiency and feasibility of the octanol to octanal conversion. Utilizing process simulation softwar, the reaction model is established, incorporating thermodynamic models to predict phase behavior and heat/mass transfer phenomena accurately. The reactor module is defined within the software, and kinetic equations and parameters are inputted. An economic analysis and evaluation are conducted to estimate costs, including raw materials, energy, equipment, and operation costs, assessing the economic feasibility and benefits under optimal operating conditions. This comprehensive approach aids engineers in making informed decisions during the design and operation phases.

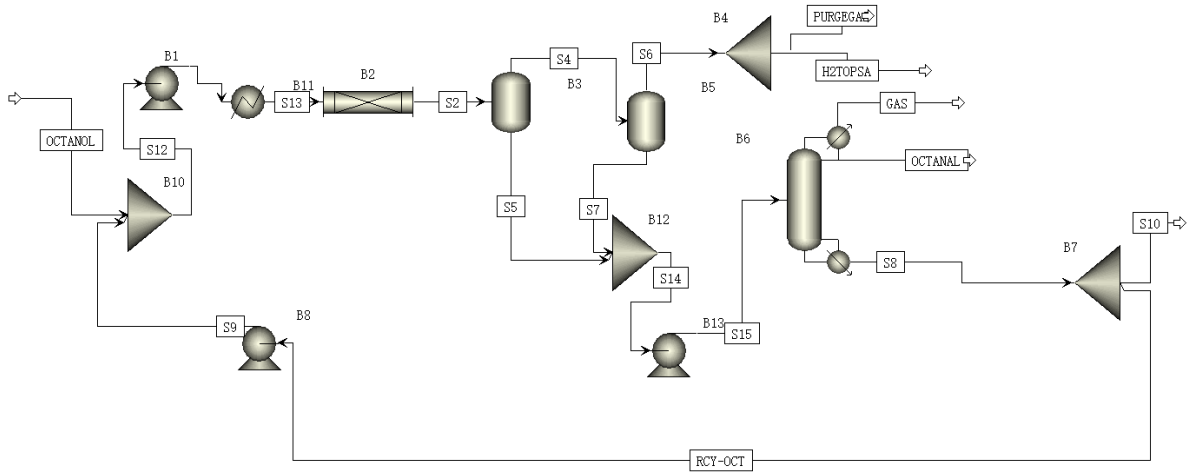

Fig. S1 Process Simulation

2. Description of DNN, DBN, XGBOOST and RF Network

Deep Neural Networks (DNNs)

Deep Neural Networks (DNNs) are a type of artificial neural network composed of multiple hidden layers, each consisting of numerous neurons. DNNs learn complex patterns and features through multiple layers of nonlinear transformations, excelling in tasks such as classification, regression, and generation. A typical DNN structure includes

an input layer, several hidden layers, and an output layer. The connections between layers are trained using the backpropagation algorithm to minimize the error function. DNNs are widely used in fields like image recognition, speech recognition, and natural language processing.

Deep Belief Networks (DBNs)

Deep Belief Networks (DBNs) are generative models composed of multiple stacked Restricted Boltzmann Machines (RBMs). DBNs are trained layer-by-layer, with each RBM learning data representations that serve as inputs for the next layer, gradually extracting high-level features of the data. Pre-trained DBNs can be used to initialize deep neural networks, addressing the gradient vanishing problem in traditional deep network training, thus improving convergence speed and performance. DBNs excel in unsupervised learning and feature extraction, commonly used for dimensionality reduction and initial weight setting.

Random Forest

Random Forest is an ensemble learning method based on decision trees. By constructing multiple decision trees, Random Forest aggregates their predictions through voting or averaging, enhancing the model's accuracy and stability. Each decision tree is trained using randomly selected features and samples, increasing model diversity and reducing the risk of overfitting. Random Forest performs well in handling high-dimensional data and complex relationships, commonly used in classification and regression tasks.

XGBoost

XGBoost (Extreme Gradient Boosting) is an efficient gradient boosting algorithm designed for fast training and high-performance prediction. It builds a series of weak learners (typically decision trees) to iteratively reduce prediction errors. XGBoost employs regularization techniques to prevent overfitting and achieves significant speed improvements through parallel computation and optimization. Due to its high performance and flexibility, XGBoost is widely used in data science competitions like Kaggle and is

suitable for various classification, regression, and ranking tasks.

3. Database, Pareto Optimization Set and Verification

The database contains **1845 data**, partial as shown in Fig S2 and Fig S3.

Table S1 The range of variables

Varible	Range	step size	Varible	Range	step size
Reaction temperature/℃	[100, 140]	1	Distillation tray numbers	[20, 40]	5
(Reaction time/h, Feed position)	([10, 12], [12, 14], [14, 16])				
Reaction pressure/MPa	0.2, 1, 1.5				

	A	B	C	D	E	F	G
1	100	0.2	10	20	12	1.455835	0.47126
2	101	0.2	10	20	12	1.450317	0.449869
3	102	0.2	10	20	12	1.44397	0.430706
4	103	0.2	10	20	12	1.43738	0.413647
5	104	0.2	10	20	12	1.430725	0.398609
6	105	0.2	10	20	12	1.42427	0.38543
7	106	0.2	10	20	12	1.418195	0.373863
8	107	0.2	10	20	12	1.412678	0.363872
9	108	0.2	10	20	12	1.40782	0.355371
10	109	0.2	10	20	12	1.40361	0.348094
11	110	0.2	10	20	12	1.400173	0.341978
12	111	0.2	10	20	12	1.397521	0.336873
13	112	0.2	10	20	12	1.395593	0.332699
14	113	0.2	10	20	12	1.394373	0.32935
15	114	0.2	10	20	12	1.393714	0.326691
16	115	0.2	10	20	12	1.393563	0.324656
17	116	0.2	10	20	12	1.393807	0.323169
18	117	0.2	10	20	12	1.394448	0.322043
19	118	0.2	10	20	12	1.395182	0.32123
20	119	0.2	10	20	12	1.395933	0.320654
21	120	0.2	10	20	12	1.396697	0.320323
22	121	0.2	10	20	12	1.39724	0.319996
23	122	0.2	10	20	12	1.397492	0.319689
24	123	0.2	10	20	12	1.397361	0.319376
25	124	0.2	10	20	12	1.396739	0.318904
26	125	0.2	10	20	12	1.395657	0.318355
27	126	0.2	10	20	12	1.393914	0.3175
28	127	0.2	10	20	12	1.391625	0.316394
29	128	0.2	10	20	12	1.388775	0.314974

data

Fig. S2 Partial Codes

A1 100

	A	B	C	D	E	F	G	H
1818	113	1.5	14	40	16	1.404828	0.326939	
1819	114	1.5	14	40	16	1.403376	0.324049	
1820	115	1.5	14	40	16	1.402299	0.321941	
1821	116	1.5	14	40	16	1.401315	0.320514	
1822	117	1.5	14	40	16	1.400591	0.319527	
1823	118	1.5	14	40	16	1.400037	0.319024	
1824	119	1.5	14	40	16	1.399603	0.318778	
1825	120	1.5	14	40	16	1.399315	0.318821	
1826	121	1.5	14	40	16	1.399072	0.318882	
1827	122	1.5	14	40	16	1.398889	0.318943	
1828	123	1.5	14	40	16	1.398634	0.31898	
1829	124	1.5	14	40	16	1.39835	0.318766	
1830	125	1.5	14	40	16	1.398093	0.318307	
1831	126	1.5	14	40	16	1.397707	0.317447	
1832	127	1.5	14	40	16	1.397197	0.316277	
1833	128	1.5	14	40	16	1.396628	0.314674	
1834	129	1.5	14	40	16	1.395899	0.312489	
1835	130	1.5	14	40	16	1.395027	0.309925	
1836	131	1.5	14	40	16	1.394071	0.306764	
1837	132	1.5	14	40	16	1.392906	0.303061	
1838	133	1.5	14	40	16	1.391568	0.298951	
1839	134	1.5	14	40	16	1.390068	0.294303	
1840	135	1.5	14	40	16	1.388347	0.289382	
1841	136	1.5	14	40	16	1.386572	0.28406	
1842	137	1.5	14	40	16	1.38463	0.278597	
1843	138	1.5	14	40	16	1.382589	0.272963	
1844	139	1.5	14	40	16	1.38039	0.267432	
1845	140	1.5	14	40	16	1.378068	0.26207	

Fig. S3 Partial Codes

For the convenience of reviewers and readers, we uploaded the database to GitHub website. xinzhouupc/DNN_DBN_RF_XGB (github.com)

4. Implementing environmental and evaluation standards

4.1 Implementing environmental

In the process of building and optimizing surrogate models, various professional software packages were leveraged. All computations were executed in a high-performance computing environment equipped with an Intel Core i7-11800H processor and 32GB of RAM. The process flow diagram was constructed in Aspen Plus V12, where the reaction mechanisms and process unit simulations were precisely executed. This resulted in a training dataset containing 620 mechanism convergence data points, providing a solid foundation for subsequent analyses. In the study, a hybrid model was developed that integrates the robust fitting capabilities of deep neural networks with the efficient search mechanisms of

multi-objective optimization algorithms. The surrogate model is constructed using deep neural networks within the Python 3.8 environment, and the multi-objective optimization algorithms, such as NSGA-II, NSGA-III, and MOEA/D from the Pymoo packages, are employed to achieve multi-dimensional optimization of the model. This approach not only improves the predictive accuracy of the model but also enhances its applicability in complex decision-making problems.

In the process of constructing the surrogate model, various machine learning algorithms were employed to achieve a multi-level construction. Specifically, three distinct surrogate modeling techniques were utilized: Deep Neural Networks (DNN), Xgboost, and Random Forest. DNNs are renowned for their powerful feature learning and representation capabilities, while Xgboost is known for its efficient gradient boosting and optimized tree structure. Random Forest, on the other hand, provides robust predictions by integrating multiple decision trees. The combined use of these algorithms aims to compare and select the optimal approach, ultimately resulting in a superior-performing and highly generalized surrogate model.

Table 1 shows the summary of surrogate modeling and optimization strategy nomenclature in this work. This table summarizes the nomenclature of the different surrogate modeling and optimization strategies used in this study. The surrogate models are categorized into the following three types: (1) Direct Solution Method (DSM): This approach utilizes COM technology to directly interact with the Aspen Plus mechanism model and optimize it through multi-objective optimization algorithms compiled in Python 3.11. This provides a direct solution to the optimization problem. (2) Deep Learning-based Surrogate Model (DLSM): This category employs algorithms such as Deep Neural Networks (DNN) and Deep Belief Networks (DBN) as surrogate models. Through training, these models capture the characteristics of complex systems and, when combined with multi-objective optimization algorithms, enable efficient optimization solutions. (3) Ensemble-based Surrogate Model (ESM): Here, Xgboost and Random Forest (RF) algorithms are utilized as surrogate models. By integrating the predictive results of multiple base learners, this approach improves the accuracy and stability of the surrogate model. Subsequently, it works synergistically with multi-objective optimization algorithms to achieve the desired optimization goals.

Table S2. The summary of surrogate modeling and optimization strategy nomenclature

Model classification	Surrogate Model and Optimization Algorithm Name	Surrogate Model Pattern	Optimization Algorithm Strategy
DSM	AP-NSGA-II	Aspen Plus	NSGA-II
	AP-NSGA-III	Aspen Plus	NSGA-III
	AP-MOEA/D	Aspen Plus	MOEA/D
	AP- C-TAEA	Aspen Plus	C-TAEA
DLSM	DNN-NSGA-II	Deep neural networks	NSGA-II
	DNN-NSGA-III	Deep neural networks	NSGA-III
	DNN-MOEA/D	Deep neural networks	MOEA/D
	DNN- C-TAEA	Deep neural networks	C-TAEA
	DBN-NSGA-II	Deep belief network	NSGA-II
	DBN-NSGA-III	Deep belief network	NSGA-III
	DBN-MOEA/D	Deep belief network	MOEA/D
	DBN- C-TAEA	Deep belief network	C-TAEA
ESM	Xgboost-NSGA-II	eXtreme Gradient Boosting	NSGA-II
	Xgboost -NSGA-III	eXtreme Gradient Boosting	NSGA-III
	Xgboost -MOEA/D	eXtreme Gradient Boosting	MOEA/D
	Xgboost - C-TAEA	eXtreme Gradient Boosting	C-TAEA
	RF-NSGA-II	Random Forest	NSGA-II
	RF-NSGA-III	Random Forest	NSGA-III
	RF-MOEA/D	Random Forest	MOEA/D
	RF- C-TAEA	Random Forest	C-TAEA

4.2 Evaluation standards

Before delving into the optimization of various surrogate modeling methods, their performance was first evaluated on both training and testing sets. To achieve this, the coefficient of determination (R^2), mean absolute percentage error (MAPE), and root mean squared error (RMSE) were employed as key evaluation metrics, aiming to comprehensively assess the model's fitting capability and prediction accuracy. The calculation equations are shown as follows:

$$R^2 = 1 - \frac{\sum_{k=1}^{K} (y_k - \hat{y}_k)^2}{\sum_{k=1}^{K} (y_k - \bar{y})^2} \tag{1}$$

$$MAPE = \frac{1}{K} \sum_{k=1}^{K} \frac{\left| y_k - \hat{y}_k \right|}{y_k} \tag{2}$$

$$RMSE = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (y_k - \hat{y}_k)^2} \tag{3}$$

To provide a precise description, K was used to denote the total number of data points, and k to represent each individual data point. Among them, the variable y_k stands for the actual output value, $\hat{y}_k$ represents the predicted output value from the surrogate model, and $\bar{y}$ denotes the average of the actual output values.

5. Verification of accuracy of surrogate model for the isooctanol oxidation process

Table S3 displays the hyperparameters for RF and XGBoost in the ensemble algorithm, while the remaining parameters mentioned adopt default values of the model. The ratio of training set and test set are 0.8:0.1:0.1.

Table S3. ESM algorithm hyperparameter settings

No.		Items	Values
1	RF	max_depth	5
2	RF	n_estimators	100
3	RF	random_state	0
4	Xgboost	learning_rate	0.05
5	Xgboost	max_depth	5
6	Xgboost	n_estimators	100

Figures S4(a) and S4(b) depict the predicted values and actual values for each target variable using the XGBoost model for training and prediction tasks, respectively. Similarly, Figures S4(c) and S4(d) illustrate the predicted values and actual values for each target variable using the RF model for training and prediction tasks, respectively. The MultiOutput Regressor method was utilized from the sklearn machine learning algorithm package to predict the multi-output ensemble algorithm. As depicted in Figure S4, the XGBoost algorithm achieved training results for predicting octanal production costs with an R^2 value of 0.998, an RMSE of 0.00085, and an MAPE of 0.00044. In contrast, the RF algorithm demonstrated superior performance in the same task, with R^2, RMSE, and MAPE values of 0.998, 0.00083, and 0.00045, respectively. Overall, both models exhibited slightly better performance on the prediction set compared to the training set, but the XGBoost model

displayed higher accuracy in predicting octanal production costs, especially in terms of the RMSE and MAPE metrics on the training set. For the prediction of unit carbon dioxide emissions, the XGBoost model also outperformed the RF model, indicating stronger generalization capabilities of the XGBoost when handling such prediction tasks.

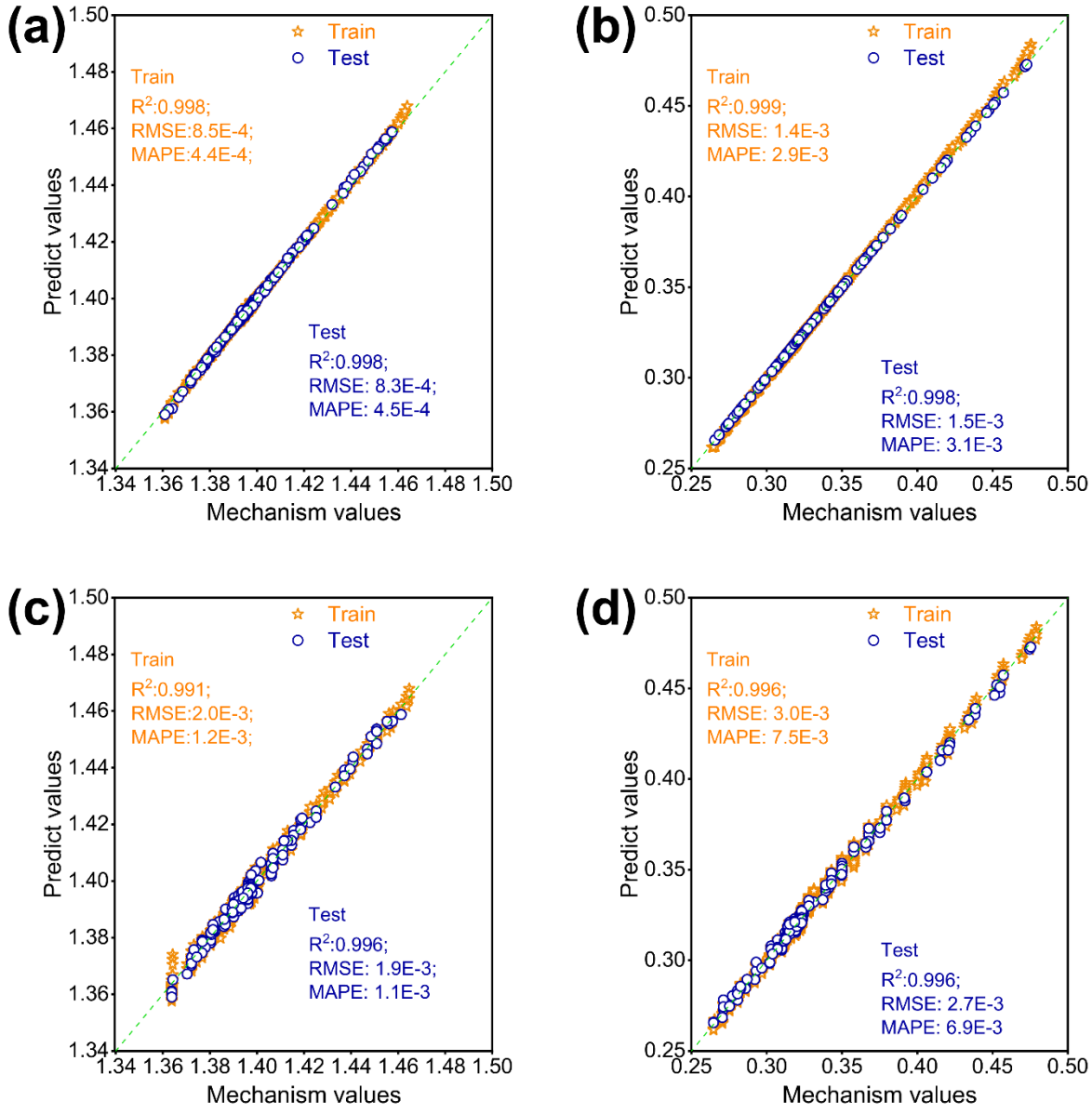

Figure S4. (a) The performance of Xgboost training and testing sets, in terms of the production cost of octanal; (b) Performance of Xgboost training and testing sets, for CO_2 unit emissions; (c) RF training set and test set performance, in terms of the production cost of octanal; (b) RF training set and test set performance, for CO_2 unit emissions

This study also focuses on the accuracy assessment of Deep Neural Networks (DNN) and Deep Belief Networks (DBN) within the Deep Learning Structure Model (DLSM) for training and prediction tasks. Through Figures S5(a) and S5(b), a detailed comparison was provided of the predicted values and actual values for each target variable during the training and prediction processes of the DNN model to evaluate its performance. Figures S5(c) and S5(d) reveal the prediction accuracy of the DBN model for the same tasks. Additionally, Table S4 presents the key hyperparameter settings for DNN and DBN within the DLSM. In this study, the ratio of training set to testing set is set to 0.9:0.1 to ensure comprehensive evaluation of model performance.

Table S4. DLSM algorithm hyperparameter settings

No.		Items	Values
1	DNN	Number of neural network layers	40
2	DNN	Number of nodes in each layer of neural network	50
3	DNN	Number of neural network iterations	300
4	DNN	Learning rate	0.001
5	DBN	Number of neural network iterations	300
6	DBN	Learning rate	0.001
7	DBN	RBM iterations	50
8	DBN	Number of neural network layers	40

This study conducted predictions of octanal production costs and unit carbon dioxide emissions for the DNN and DBN models within the DLSM, showcasing the predictive results of both algorithms. As depicted in Figure S5, the DNN algorithm achieved training results for predicting octanal production costs with an R^2 value of 0.997, an RMSE of 0.0011, and an

MAPE of 0.00061. In contrast, the DBN algorithm demonstrated superior performance in the same task, with R^2, RMSE, and MAPE values of 0.999, 0.00041, and 0.00022, respectively. Overall, both models exhibited slightly better performance on the prediction set compared to the training set, but the DBN model displayed higher accuracy in predicting octanal production costs, especially in terms of the RMSE and MAPE metrics on the training set. For the prediction of unit carbon dioxide emissions, the DBN model also outperformed the DNN model, indicating stronger generalization capabilities of the DBN when handling such prediction tasks.

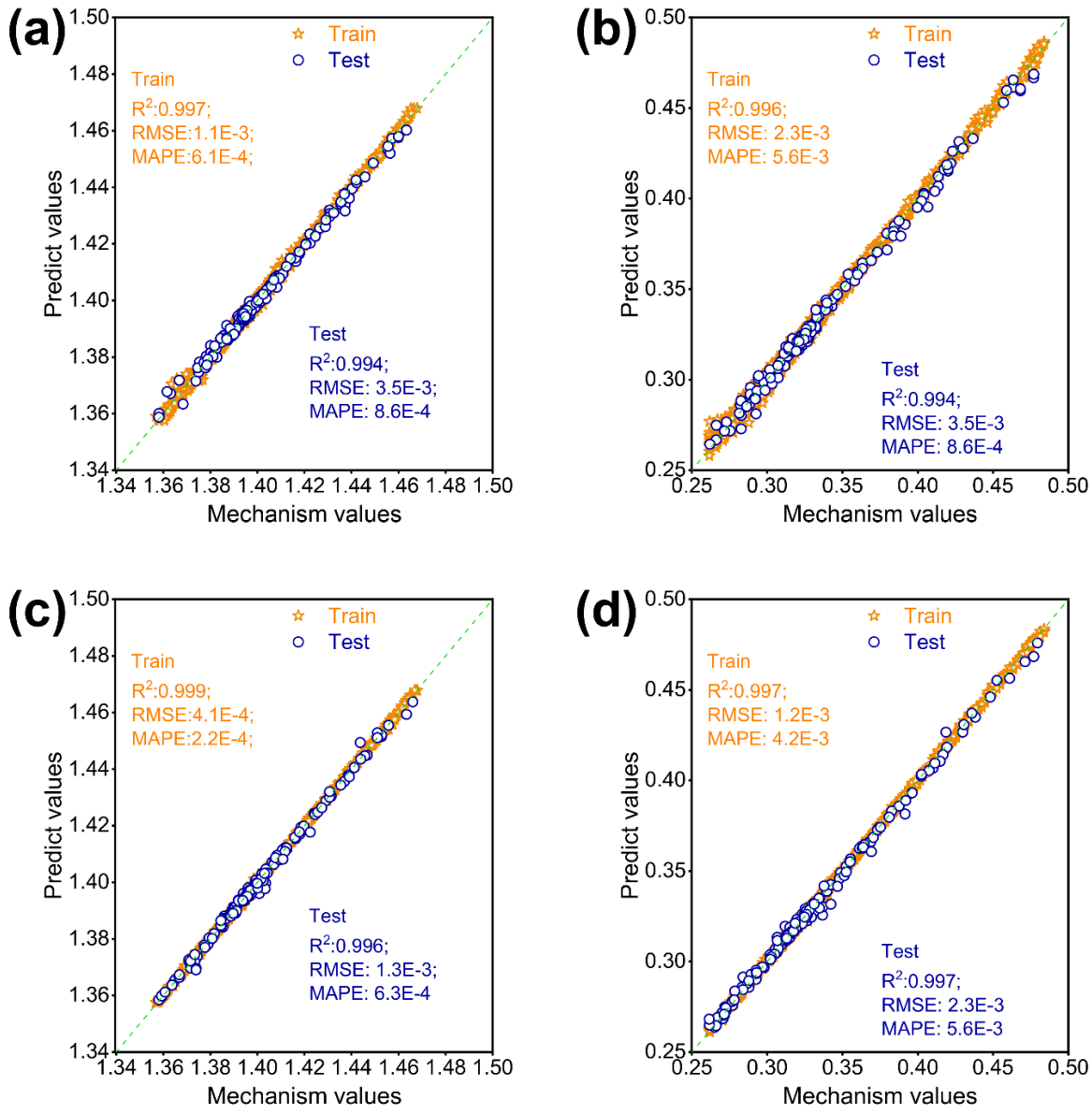

Figure S5. (a) The performance of DNN training and testing sets, in terms of the production cost of octanal; (b) Performance of DNN training and testing sets, for CO_2 unit emissions; (c) DBN training set and test set performance, in terms of the production cost of octanal; (b) DBN training set and test set performance, for CO_2 unit emissions

6. Discussions on model convergence

Ensuring iterative convergence is crucial for guaranteeing the accuracy and efficiency of simulations. It is essential to select appropriate methods and strategies based on the specific circumstances. Specifically, when faced with non-convergence issues during simulation, the following measures can be taken:

(a) When encountering convergence issues, the first step is to check whether the data settings are reasonable. If there are no issues with the data settings themselves, making slight adjustments to the relevant parameters within a reasonable range can often effectively avoid convergence problems.

(b) Reassess the appropriateness of the selected property methods. The choice of property methods should consider several factors, including the type of fluid, required accuracy, computational cost, and the availability of property data.

(c) Select an appropriate iteration algorithm. Aspen offers a range of iteration algorithms, each designed for specific applications. It is essential to review and choose the default convergence method or select other more suitable iteration algorithms based on the specific circumstances.

(d) Adjust the iteration control strategy based on the findings in (b). For instance, if convergence issues arise, it may be due to an insufficient number of iterations. In such cases, consider increasing the maximum number of iterations for the convergence method. However, it is crucial to minimize the number of iterations required while still meeting the specified convergence criteria, thereby conserving computational resources and time.

7. Verified results

This study employed the Aspen Plus software platform to systematically validate the solutions obtained from surrogate-based optimization. After a comprehensive evaluation of various combinations of surrogate models and optimization strategies during the preliminary phase, the XGBoost-RVEA integrated model was selected as the core analytical tool for in-depth investigation. The optimized solutions generated by this model

were subsequently imported into Aspen Plus for full-process simulation verification, with the corresponding validation results detailed in Table S5.

Table S5 XGBoost-RVEA model validation results

		Surrogate model	Simulation	relative error
Case 1	production cost	1.29564	1.29579	-0.01158
	CO_2 emission	0.28924	0.28935	-0.03802
Case 2	production cost	1.30303	1.30415	-0.08588
	CO_2 emission	0.28488	0.28512	-0.08418
Case 3	production cost	1.31043	1.31096	-0.04043
	CO_2 emission	0.28081	0.28109	-0.09961
Case 4	production cost	1.31783	1.31688	0.07214
	CO_2 emission	0.27704	0.2769	0.05056
Case 5	production cost	1.32523	1.32537	-0.01056
	CO_2 emission	0.27356	0.27374	-0.06576
Case 6	production cost	1.33263	1.33196	0.050302
	CO_2 emission	0.27039	0.27026	0.048102
Case 7	production cost	1.34003	1.33965	0.028366
	CO_2 emission	0.26751	0.26726	0.093542
Case 8	production cost	1.34743	1.34652	0.067582
	CO_2 emission	0.26493	0.26488	0.018876
Case 9	production cost	1.35483	1.35522	-0.02878
	CO_2 emission	0.26264	0.26284	-0.07609
Case 10	production cost	1.36223	1.36258	-0.02569
	CO_2 emission	0.26066	0.26084	-0.06901

8. Multi-objective optimization parameters

The parameter configurations of the four multi-objective evolutionary algorithms used in this study are shown in Table S6.

Table S6 The parameters for multi-objective algorithms.

Parameters	NSGA-II	NSGA-III	MOEA/D	RVEA
pop_size	40	40	——	——
n_partitions	——	10	12	12
n_gen	300	100	200	400
n_neighbors	——	——	15	——
prob_neighbor_mating	——	——	0.7	——

References

[1] Shi R, Wang F, Mu X, et al. Transfer dehydrogenation of 1-octanol to 1-octanal over cu/mgo catalyst: Effect of cu particle size [J]. Chinese Journal of Catalysis, 2010, 31(6): 626-630.

[2] Fei Wang, Ruijuan Shi, Zhi-Quan Liu, et al., Highly Efficient Dehydrogenation of Primary Aliphatic Alcohols Catalyzed by Cu Nanoparticles Dispersed on Rod-shaped La2O2CO3, ACS Catalysis 2013, 3, 5, 890-894.