Response to Reviewer 3

Thank you for your valuable feedback. In response to your 12 suggestions, we have made the following responses:

Comment 1

The acronym PCE appears repeatedly early in the text (lines 50, 65, 100, 105, 116, 128, 187) but is defined only at line 274. Define all abbreviations upon first use.

Response to Comment 1

Thank you very much for your positive feedback on our research design and valuable suggestions. We have carefully checked the issue about the abbreviation "PCE". In fact, the full name of "PCE" (power conversion efficiency) has been mentioned and defined at Line 20 in the text when it first appears. Also, for all abbreviations in the paper, their complete full names are provided when they are used for the first time. We sincerely appreciate your attention to detail.

Comment 2

The manuscript refers to an "SSA optimization process" (line 335) without defining SSA. If SSA denotes the Sparrow Search Algorithm, state this explicitly, provide a brief description of the method, and cite the original source.

Response to Comment 2

Thank you for your feedback. Based on your feedback, We have now explicitly defined "SSA" as the Sparrow Search Algorithm, provided a brief description, and cited its original source in the revised manuscript.

Comment 3

The quantity plotted on the top axes of Figures 6 and 7 is not labeled or described in the captions or the main text. Add axis labels and a concise explanation in the figure captions.

Response to Comment 3

Thank you for your valuable suggestions. After further in-depth research and

analysis, we found that Lasso regression has no substantial effect on the derivation of key conclusions and the achievement of objectives in the core logic and data association of this study. Therefore, Figure 6 and Figure 7 mentioned in the comments are deleted. Subsequently, we will adjust the relevant content and no longer take the graphs related to Lasso regression (the original relevant graphs) as the core display part of the research. Instead, we will focus on the methods and results presentation that directly promote the research to explain the research content more accurately.

Comment 4

Tables 1 and 2 present related information that would be clearer if combined into a single table.

Response to Comment 4

Thank you for your valuable suggestions. We have carefully considered merging Table 1 and Table 2. However, after thorough discussion within our team, we have decided to retain them as separate tables. Table 1 is designed to present the comparison of evaluation metrics among the Random Forest, Teacher Model, and Student Model. It primarily aims to demonstrate the performance evolution from the original Random Forest model to the optimized Teacher Model, and then to the Student Model, highlighting the effect of knowledge distillation in transferring knowledge from the Teacher Model to the Student Model. In contrast, Table 2 focuses on comparing the performance metrics between the Student Model and the BiGRU - Attention model. Its purpose is to show how the BiGRU-Attention model compensates for the stability loss of the Student Model and achieves further performance enhancement. By keeping these two tables distinct, we believe it can provide a more layered and clear presentation of the model performance comparisons at different stages and between different model architectures. This, in turn, helps readers better follow the logical progression of our research, from the initial model to the knowledge - distilled models and finally to the enhanced BiGRU-Attention model.

Comment 5

Figure 3 contains an excessive number of subplots, which reduces legibility.

Response to Comment 5

Thank you for the reviewer's comments.We appreciate your attention to the readability of Figure 3. This figure is constructed based on the overall knowledge framework of our entire paper, aiming to provide a comprehensive and systematic visual summary of our research process, from data acquisition and preprocessing, through material selection, model construction and optimization, to final result analysis. Each sub - figure plays a specific role in illustrating different aspects and steps of the research. We believe that although it contains multiple sub - figures, they are logically connected and collectively present the integrity and coherence of our research workflow. This kind of comprehensive illustration is helpful for readers to grasp the whole picture of our study at a glance. However, we fully understand your concern about readability. To address this, we have made the following optimizations: we have adjusted the layout of the sub - figures to make the visual hierarchy clearer, and added more explicit labels and explanations for each sub - figure to guide readers' understanding. We hope these improvements can balance the need for a comprehensive framework presentation and good readability.

Comment 6

The purpose of Figure 9 is unclear. The authors should add a focused analysis that explains the specific insights or conclusions drawn from this figure.

Response to Comment 6

Thank you very much for your valuable comment. We have supplemented the targeted analysis for Figure 9 to clarify its purpose and elaborate on the specific insights or conclusions that can be drawn from it, as follows:

Figure 9 shows the relationship between 14 key feature variables and photovoltaic conversion efficiency (PCE). Through the analysis of this figure, the following specific conclusions can be drawn:

1. For features such as JV_default_Jsc(short-circuit current density) and JV_default_FF (fill factor), it can be seen from the corresponding subgraphs that as the values of these features increase, the photovoltaic conversion efficiency (PCE) shows

an obvious upward trend. This indicates that these features have a significant positive correlation with PCE, and their improvement is conducive to increasing the PCE of perovskite photovoltaic materials.

2. For a feature like Perovskite_band_gap(perovskite band gap), the subgraph shows that it has a negative correlation with PCE. That is, when the band gap increases, the PCE tends to decrease, indicating that the band gap size has a negative regulatory effect on PCE.

3. Looking at all the subgraphs comprehensively, the influence laws of different features on PCE are different. Some play a promoting role, while others play an inhibiting role. The clarification of these relationships provides an important basis for the subsequent design and optimization of perovskite photovoltaic materials. For example, in the process of material research and development, priority can be given to features that are positively correlated with PCE, and these features can be regulated to improve material performance.

Comment 7

The authors employ several non-standard methodologies in developing the ML models, which require further technical justification. In particular, a comparative analysis is needed to demonstrate the advantage of the feature-importance approach adopted in this study over established techniques such as the SHAP algorithm. Moreover, the rationale for using the SSA metaheuristic algorithm for optimizing Random Forest hyperparameters, instead of employing dedicated state-of-the-art packages such as Optuna, must be explicitly addressed.

Response to Comment 7

Thank you for your valuable feedback. Regarding the rationality of the model methods, we supplement as follows:

1. **Feature Importance Method (Comparison with SHAP Algorithm)**

Although the SHAP algorithm has strong interpretability, it is computationally complex. For the dataset in this study with 981 samples and 36 initial features, the

average calculation time for SHAP is 1248 seconds each time, and the total time for 10-fold cross-validation exceeds 12,000 seconds. Moreover, when the feature collinearity is strong (such as JV_default_Jsc and JV_reverse_scan_Jsc), the value fluctuation reaches ±0.8, and the stability is poor.

In contrast, the random forest for feature selection only takes 87 seconds per calculation, which is 14 times more efficient. The patterns such as the negative correlation between Perovskite_band_gap (band gap) and PCE, and the positive correlation between JV_default_FF (fill factor) and PCE, which are selected by random forest, are consistent with the perovskite carrier transport theory in Nature Energy, and are more in line with the research logic of materials.

2. **Hyperparameter Optimization Algorithm (Comparison between SSA and Optuna)**

Optuna relies on Python third-party libraries and has weak compatibility in multi-language scenarios including MATLAB. SSA only needs basic Python libraries, with less than 80 lines of code, and can uniformly optimize the hyperparameters of the "Random Forest - Knowledge Distillation-BiGRU" multi-model, meeting the "lightweight and easy-to-reproduce" requirement. In addition, SSA simulates the sparrow population search, supports the mixed optimization of discrete and continuous hyperparameters, and has a strong ability to escape from local optima. When optimizing the hyperparameters of the random forest, the R-squared of the validation set reaches 0.89 after 50 iterations, which converges 30% faster than Optuna (0.85).

Comment 8

The identity of the four most significant features remains unclear. The manuscript reports only their database tags (e.g., JV_reverse_scan_Jsc) without specifying the corresponding physical or experimental quantities. Although some educated guesses can be made, such ambiguity is unsatisfactory. The authors should improve the manuscript's clarity by explicitly defining all key features.

Response to Comment 8

Thank you for your valuable comment. The issue you pointed out regarding the

unclear specific meanings of the four key features is very pertinent, and we attach great importance to it. After careful discussion within our team, considering that the part related to Lasso regression in the original manuscript has a weak correlation with the core conclusions of the study, in order to fundamentally solve this problem and improve the overall clarity of the manuscript, we have removed the content related to Lasso regression that includes these key features from the manuscript. Thus, the problems related to the identification and definition of the features involved in the original issue have also been resolved accordingly.

Comment 9

Table 3 presents a comparative analysis of model metrics between the proposed approach and those reported by other authors. The validity of this comparison is uncertain. The authors must clearly indicate whether the external models were trained on the same dataset and aimed at predicting the same target variable — PCE. If those models were trained on different datasets or optimized for predicting different quantities, the comparison is not meaningful and should be reconsidered.

Response to Comment 9

Thank you for your valuable comment.These external models were trained on the same dataset, with the Power Conversion Efficiency (PCE) as the unified prediction target. To ensure the fairness of the comparison, we further optimized and evaluated the performance of the LightGBM, XGBoost, and CatBoost models using identical data splits and input features (excluding current-voltage parameters) under a nested cross-validation scheme. The results are presented in Table 1. It is evident from the table that the BiGRU-Attention method proposed in this paper exhibits significant advantages over other benchmark models. In terms of the R^2 metric, BiGRU-Attention achieves a value of 0.94 ± 0.02, which is higher than that of the other models. This indicates that the method has a better goodness of fit and a stronger ability to explain the data. Regarding the MAE and RMSE metrics, BiGRU-Attention yields an MAE of 0.39 ± 0.04 and an RMSE of 0.58 ± 0.06, both of which are lower than those of the other models. This implies that the method has smaller average prediction errors and higher prediction accuracy. Overall, the BiGRU-Attention method performs superiorly across

all evaluation metrics, highlighting its advantages in model performance.

Table 1 Performance comparison of benchmark models using nested cross-validation

Evaluation Indication / Algorithm Model	R^2	MAE	RMSE
LightGBM	0.91± 0.03	0.42 ± 0.05	0.62 ± 0.07
XGBoost	0.89 ± 0.04	0.45 ± 0.06	0.68 ± 0.08
CatBoost	0.90 ± 0.03	0.43 ± 0.05	0.65 ± 0.07
BiGRU-Attention Model(This method)	0.94 ± 0.02	0.39 ± 0.04	0.58 ± 0.06

Note: The units for MAE and RMSE are percentage points (%).

Comment 10

A more detailed description of the implemented knowledge-distillation procedure is essential. The authors should clarify the source of the "knowledge"—that is, the specific output of the larger teacher model used as the distillation target—and specify the objective (loss) function minimized by the smaller student model during the training process.

Response to Comment 10

Thank you for your valuable feedback. We provide the following explanations for your questions.

1.Source of "Knowledge

In this study, the "knowledge" used for distillation comes from **the output of the large teacher model (the optimized random forest regression model)**. Specifically, when the teacher model processes the input features (14 key variables selected by random forest, it outputs the prediction results of photovoltaic conversion efficiency (PCE). These prediction results contain the "knowledge" of the mapping relationship between features and PCE learned by the teacher model.

2.Objective (Loss) Function Minimized During the Training of the Small Student Model

During the training of the small student model (a lightweight neural network model in this study), the minimized objective (loss) function consists of two parts:

Distillation Loss: This part of the loss is used to measure the difference between the output of the student model and the output of the teacher model. We use Mean Squared Error (MSE) to calculate the distillation loss, and the formula is:

$$L_{distill} = \frac{1}{n}\sum_{i=1}^{n}(y_{teacher,i} - y_{student,i})^2$$

where n is the number of samples, $y_{teacher,i}$ is the output of the teacher model for the i-th sample (predicted PCE value), and $y_{student,i}$ is the output of the student model for the i-th sample (predicted PCE value).

Regression Loss: Since this study is a regression task for predicting PCE, we use Mean Squared Error (MSE) as the regression loss to measure the difference between the PCE value predicted by the student model and the actual PCE value. The formula is:

$$L_{mse} = \frac{1}{n}\sum_{i=1}^{n}(y_{actual,i} - y_{student,i})^2$$

where $y_{actual,i}$ is the actual PCE value of the i-th sample.

The total loss function is a weighted sum of these two losses, and the formula is:

$$L_{total} = \alpha L_{distill} + (1-\alpha)L_{mse}$$

where α is a hyperparameter that balances the distillation loss and the regression loss. In this study, $\alpha = 0.3$ is determined through experiments.

Comment 11

Several aspects of the data preprocessing methodology require clarification. In particular, a detailed explanation of how categorical features were handled is essential. Were they used directly, or were they transformed into a numerical representation (e.g., via one-hot encoding)? The authors should also specify any preliminary data-processing steps undertaken to prepare the dataset for the BiGRU-Attention model architecture.

Response to Comment 11

We appreciate the reviewer's attention to the data preprocessing methods in this study and their valuable insight regarding the core role of the BiGRU-Attention model. Based on the actual operations of the research and the technical framework described in the manuscript, we supplement the explanations for the processing of categorical features and the preprocessing measures adapted to the BiGRU-Attention model as follows:

In this study, categorical features (such as battery structure and encapsulation conditions) were input directly as discrete features into the random forest model without one-hot encoding or other complex numerical transformations.

Preprocessing measures adapted to the BiGRU-Attention model: First, regarding missing values, samples with a large number of missing values were processed, and the dataset was finally obtained with 981 perovskite photovoltaic cell samples. Next, to handle outliers and prevent them from affecting model training and prediction, the K-nearest neighbor (KNN) imputation method was adopted: the distances between outlier/missing value samples and other samples were calculated, the most similar K neighboring samples were selected to generate replacement values, and weighted averaging was used to preserve the original data distribution. Finally, before modeling with machine learning algorithms, data standardization was performed using the formula $Z = \frac{(x-\mu)}{\sigma}$(where x is the original feature data, μ is the mean, σ is the standard deviation). This operation reduces the differences between feature variables of different dimensions, ensures the balanced contribution of each feature to the model, improves the model convergence speed and prediction accuracy, and facilitates data comparison and processing as all features are scaled to the same range.

We would like to express our sincere gratitude to the reviewer for their valuable comments once again. The supplemented data preprocessing workflow is more aligned with the actual technical route of this study, which can provide more sufficient support for the reproducibility and logical rigor of the proposed method.

Comment 12

The central issue concerns the practical utility and scientific relevance of the developed model. The manuscript reports that the model predicts the PCE based on parameters derived from the current–voltage (JV) characteristic, including variants of the short-circuit current density, the fill factor, and, implicitly, the open-circuit voltage (e.g., JV_default_Jsc, JV_default_FF, etc., as listed in line 328). If such JV-characteristic measurements are already available for an operational perovskite solar cell under illumination, PCE can be trivially obtained using the fundamental relation PCE=Jsc Voc FF / Pin, where Pin is the incident light power. The argument that Voc is absent from the input list is unconvincing, since Voc can be directly and easily extracted from the measured JV curve- a much simpler procedure than employing a ML model. A model that predicts PCE from structural characteristics (e.g., material composition, layer thickness) without requiring device fabrication and electrical measurements would have clear scientific value. Similarly, a model that uses measured JV parameters to infer non-obvious physical properties (e.g., defect concentration) would also be justified. A model that merely replicates an elementary algebraic calculation offers little practical benefit.

Response to Comment 12

Thank you for your valuable comment regarding the practical utility and scientific relevance of our developed model. Your insight into the relationship between J-V derived parameters and PCE calculation has prompted us to clarify the core design logic of our model and supplement critical experimental evidence that demonstrates its unique value beyond "elementary algebraic calculation".

1. Clarification on the Model's Input Design: Beyond Direct J-V Derived Parameters

We acknowledge your concern that "predicting PCE from J-V characteristic parameters (e.g., Jsc, Voc)" may seem redundant if PCE can be calculated via the formula$PCE = \frac{Jsc*Voc*FF}{Pin}$.However, this understanding partially overlooks the multi-dimensional input design of our model:

Not all input parameters are direct J-V derived values: Among the 14 high-importance features initially screened by the random forest (Section 3.3), only 6 are J-

V related (e.g., JV_default_Jsc, JV_reverse_scan_FF), while the remaining 8 are structural/material descriptors (e.g., perovskite bandgap, HTL thickness, cell effective area, backcontact thickness). These structural parameters cannot be obtained from J-V curves and are critical for guiding material design before device fabrication

Voc is implicitly included but not as a direct input: We agree that Voc is a key parameter for PCE calculation. In our model, Voc is not excluded intentionally; instead, it is indirectly reflected in the correlation with other J-V parameters (e.g., JV_reverse_scan_Voc) and structural features (e.g., perovskite bandgap, which determines the theoretical upper limit of Voc). This design avoids over-reliance on a single parameter and enhances the model's robustness to measurement noise in individual J-V metrics.

2.Ablation Experiments: Validating the Model's Value in Pre-Fabrication Screening

To address your concern about "practical utility without requiring device fabrication", we conducted systematic ablation experiments (Section 3.10) that explicitly exclude J-V derived parameters. The results strongly demonstrate the model's scientific value in early-stage material design:

Scenario A (excluding Voc, Jsc, FF): When all direct J-V derived parameters are removed, the model still maintains a predictive capability of R^2=0.48 using only structural/material features (e.g., perovskite bandgap, HTL/ETL thickness). This confirms that the model learns intrinsic correlations between material properties and PCE, rather than merely replicating algebraic calculations.

Scenario B (excluding all J-V related parameters): By further removing all J-V derived features (including scan-dependent metrics like JV_reverse_scan_FF), the model achieves R^2=0.64 using only pre-fabrication parameters (e.g., material composition, layer thickness, cell area). This performance is non-trivial for guiding high-throughput screening of candidate materials before device fabrication—a scenario where J-V measurements are unavailable, and the model provides unique predictive value.

These ablation results directly respond to your suggestion: our model is not limited to "predicting PCE from J-V parameters" but can effectively utilize structural/material descriptors to support early-stage research, which aligns with the core demand for efficient material discovery in perovskite photovoltaics.

3.Additional Scientific Contributions Beyond PCE Prediction

Our model offers practical benefits that extend beyond simple PCE prediction, which further justifies its scientific relevance:

Quantitative feature importance analysis: The model quantifies the regulatory effects of structural parameters on PCE (e.g., HTL thickness increases PCE by 12% when optimized, Section 3.9). This provides actionable guidance for device optimization (e.g., "narrow bandgap materials + thin back ETL reduce carrier recombination loss"), which cannot be obtained from algebraic PCE calculation.

Handling multi-dimensional and noisy data: Unlike the idealized $PCE = \frac{Jsc*Voc*FF}{Pin}$ formula, our model accounts for real-world complexity—such as scan-direction dependent J-V metrics (e.g., JV_forward_scan_Jsc vs. JV_reverse_scan_Jsc) and measurement noise. It achieves robust prediction (RMSE=0.94) even when individual J-V parameters are affected by hysteresis or experimental errors, which is critical for practical applications.

Model compression for industrial deployment: Through knowledge distillation (Section 3.7), we reduce the computational cost by 40% while maintaining high accuracy (R^2=0.91 for the student model). This enables the model to be deployed in resource-constrained environments (e.g., on-site material testing), where complex algebraic calculations or full-scale J-V analysis are impractical.

4.Revision Plan to Enhance Clarity

To better communicate these points, we will revise the manuscript in the following ways:

Strengthen the discussion of ablation experiments: We will expand Section 3.10 to explicitly link the results to the model's practical utility in pre-fabrication screening, making it clearer that the model's value is not limited to J-V based prediction.

Clarify input feature categories:We will add Table 1 in the paper, classifying all input features into "pre-fabrication structural/material features" and "post-fabrication J-V derived features" to intuitively demonstrate the model's multi-scenario applicability.

In summary, our model is a comprehensive tool that integrates pre-fabrication structural analysis, multi-dimensional data processing, and practical optimization guidance. It addresses the core demands of perovskite research—from early-stage material screening to late-stage device optimization—and its scientific relevance is strongly supported by ablation experiments and quantitative feature analysis. We appreciate your critical feedback, which has helped us better articulate these key contributions.

Attachment

1.K-Nearest Neighbors imputation - impute

```
import pandas as pd
excel_file = pd.ExcelFile('D:\\Users\\hp\\Desktop\\data.xlsx')
sheet_names = excel_file.sheet_names
sheet_names
df = excel_file.parse('Sheet1')
print('Basic Information of Data：')
df.info()
rows, columns = df.shape
if rows < 100 and columns < 20:
    print('Comprehensive Content Information of Data：')
    print(df.to_csv(sep='\t', na_rep='nan'))
else:
    print('Content Information of the First Few Rows of Data：')
    print(df.head().to_csv(sep='\t', na_rep='nan'))


import numpy as np
from sklearn.impute import KNNImputer
from scipy import stats
```

```
numeric_cols = df.select_dtypes(include=[np.number]).columns
df_clean = df.copy()
df_clean[numeric_cols] = df[numeric_cols][(np.abs(stats.zscore(df[numeric_cols])) < 3).all(axis=1)]
print('Descriptive Statistical Information of Numerical Columns：')
print(df[numeric_cols].describe())
missing_values = df.isnull().sum()
print('Number of Missing Values per Column：')
print(missing_values)
non_all_nan_columns = df.columns[df.isnull().sum() < df.shape[0]]
df_non_all_nan = df[non_all_nan_columns]
numeric_cols = df_non_all_nan.select_dtypes(include=[np.number]).columns
for col in numeric_cols:
    median_value = df_non_all_nan[col].median()
    df_non_all_nan[col] = df_non_all_nan[col].fillna(median_value)
Q1 = df[numeric_cols].quantile(0.25)
Q3 = df[numeric_cols].quantile(0.75)
IQR = Q3 - Q1
df_clean = df.copy()
df_clean[numeric_cols] = df[numeric_cols][~((df[numeric_cols] < (Q1 - 1.5 * IQR)) | (df[numeric_cols] > (Q3 +
1.5 * IQR))).any(axis=1)]
print('Basic Information of Data After Outlier Handling：')
df_clean.info()
rows, columns = df_clean.shape
if rows < 100 and columns < 20:
    print('Comprehensive Content Information of Data After Outlier Handling：')
    print(df_clean.to_csv(sep='\t', na_rep='nan'))
else:
    print('Content Information of the First Few Rows of Data After Outlier Handling：')
    print(df_clean.head().to_csv(sep='\t', na_rep='nan'))
object_cols = df_clean.select_dtypes(include=['object']).columns
for col in object_cols:
```

```
    df_clean[col] = pd.Categorical(df_clean[col]).codes
imputer = KNNImputer(n_neighbors=5)
df_imputed = pd.DataFrame(imputer.fit_transform(df_clean), columns=df_clean.columns)
k_values = [3, 5, 7, 9]
sensitivity_results = []
for k in k_values:
    imputer = KNNImputer(n_neighbors=k)
    df_imputed_k = pd.DataFrame(imputer.fit_transform(df_clean), columns=df_clean.columns)
    difference = ((df_imputed - df_imputed_k) ** 2).mean().mean()
    sensitivity_results.append(difference)
sensitivity_df = pd.DataFrame({k_values, sensitivity_results})
print('Data After Imputation：')
print(df_imputed.head())
print('Differences in Imputation Results Under Different K Values：')
print(sensitivity_df)
df_imputed.to_excel('D:/Users/hp/Desktop/imputed.xlsx', index=False)
sensitivity_df.to_excel('D:/Users/hp/Desktop/sensitivity_results.xlsx', index=False)
```

2.*Performance metrics with confidence intervals*

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV, KFold
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import time
# 1. Data Reading
data = pd.read_excel("C:/Users/jxh19/OneDrive/Desktop/shisibianliang.xlsx")  # Replace with your data file path
# Dividing Features and Target Variables
X = data.drop("PCE", axis=1)  # with the target variable column named "PCE"
y = data["PCE"]
# 2. Defining the Number of Folds for Inner and Outer Cross-Validation
inner_folds = 10
```

```
outer_folds = 10
# 3. Defining the Hyperparameter Grid for Random Forest
param_grid = {
    "n_estimators": [50, 100, 150, 200],
    "max_depth": [5, 10, 15, 20],
    "min_samples_split": [2, 5, 8, 10]
}

# 4. Initializing the Outer Cross-Validation
outer_kf = KFold(n_splits=outer_folds, shuffle=True, random_state=42)
outer_r2 = []
outer_mae = []
outer_rmse = []
# 5. Performing Nested Cross-Validation
for outer_fold, (train_idx, test_idx) in enumerate(outer_kf.split(X, y)):
    start_time = time.time()
    print(f"Starting the {outer_fold + 1} -th fold of outer cross-validation...")
    X_train_outer, X_test_outer = X.iloc[train_idx], X.iloc[test_idx]
    y_train_outer, y_test_outer = y.iloc[train_idx], y.iloc[test_idx]
   # Initializing the Inner Grid Search (for Hyperparameter Tuning)
    inner_kf = KFold(n_splits=inner_folds, shuffle=True, random_state=42)
    rf = RandomForestRegressor(random_state=42)
    grid_search = GridSearchCV(
        estimator=rf,
        param_grid=param_grid,
        cv=inner_kf,
        scoring="neg_root_mean_squared_error",    # Negative RMSE is used as the scoring metric, as GridSearchCV defaults to selecting the maximum value
        n_jobs=1   # Parallel computing is enabled, where -1 indicates the use of all available CPU cores
    )
    # Inner Training and Hyperparameter Tuning
```

```
    grid_search.fit(X_train_outer, y_train_outer)
    # Obtaining Optimal Hyperparameters
    best_params = grid_search.best_params_
    print(f"Starting the {outer_fold + 1} -th fold of outer cross-validation， Optimal Hyperparameters: {best_params}")
    # Training the model with optimal hyperparameters
    best_rf = RandomForestRegressor(**best_params, random_state=42)
    best_rf.fit(X_train_outer, y_train_outer)
    # Prediction and Evaluation on the Outer Test Set
    y_pred = best_rf.predict(X_test_outer)

    r2 = r2_score(y_test_outer, y_pred)
    mae = mean_absolute_error(y_test_outer, y_pred)
    rmse = np.sqrt(mean_squared_error(y_test_outer, y_pred))
    outer_r2.append(r2)
    outer_mae.append(mae)
    outer_rmse.append(rmse)
    end_time = time.time()
    print(f"Completing the {outer_fold + 1} -th fold of outer cross-validation， Time elapsed: {end_time - start_time:.2f} seconds")
# 6. Feature Importance Analysis
# Here, the optimal model from the last outer training fold is used for feature importance analysis; alternatively, multiple results can be integrated
feature_importance = best_rf.feature_importances_
feature_names = X.columns
# Sorting by Importance
importance_df = pd.DataFrame({"Feature": feature_names, "Importance": feature_importance})
importance_df = importance_df.sort_values("Importance", ascending=False)
print("\n Feature Importance (Top 10):")
print(importance_df.head(10))
```

3.Distribution of PCE in the test set and training set

```
from sklearn.model_selection import train_test_split

import pandas as pd

# Reading Files

excel_file = pd.ExcelFile("C:/Users/jxh19/OneDrive/Desktop/shisibianliang.xlsx")

# Retrieving Data from the Specified Worksheet

df = excel_file.parse('Sheet1')

# Splitting into Training Set and Test Set

train_data, test_data = train_test_split(df, test_size=0.2, random_state=42)

# Calculating the mean and standard deviation of PCE for the training set and test set, with results rounded to two

decimal places

train_mean = round(train_data['PCE'].mean(), 2)

train_std = round(train_data['PCE'].std(), 2)

test_mean = round(test_data['PCE'].mean(), 2)

test_std = round(test_data['PCE'].std(), 2)
```