Supporting Information

Machine Learning Optimization of P-Type Transparent Conducting Films

Lingfei Wei,[†,‡] *Xiaojie Xu,*[§] *Gurudayal,*[‡,⊥] *James Bullock,*[‖] *and Joel W. Ager,*[*,‡,⊥,∇]

[†] School of Chemistry and Chemical Engineering, Southeast University, Nanjing 211189, P. R. China

[‡] Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America

[§] Materials Sciences Division, Lawrence Livermore National Laboratory, Livermore, CA 94550, United States of America

[⊥] Joint Center for Artificial Photosynthesis, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America

[‖] Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Victoria, Australia

[∇] Department of Materials Science and Engineering, University of California Berkeley, Berkeley, CA 94720, United States of America

Table of contents

Home-built thermopower system setup details[1,2]

Thermopower measurements were performed by measuring the voltage that develops across a sample when a temperature gradient is applied, shown schematically in Figure S1. The temperature gradient is performed by suspending samples between two temperature-controlled copper blocks. With voltage applied and the contacted cold finger filled with liquid nitrogen, temperature at both sides can be controlled independently by resistive heating and liquid nitrogen cooling. Lakeshore 331 temperature controllers using Proportional/Integral/Derivative (PID) algorithms are employed for precise, automated control of temperature, which supply the current to the resistive heaters and use the thermocouple inputs as control sensors. Platinum RTD (resistive temperature device) sensors are also attached to the bottom of each sample block and read by the temperature controllers as secondary inputs. The Seebeck voltage is then measured between the thermocouples using Keithley 2000-20 multimeter. The sample stage and cold finger are enclosed inside a small vacuum chamber connected to a small turbo pump and backed by a mechanical roughing pump providing pressures as low as 9 × 10^{-7} torr. The vacuum environment provides excellent thermal insulation and prevents substances in the air (such as water vapor) from condensing on the sample and wiring inside when conducting measurements at low temperatures.

What ?

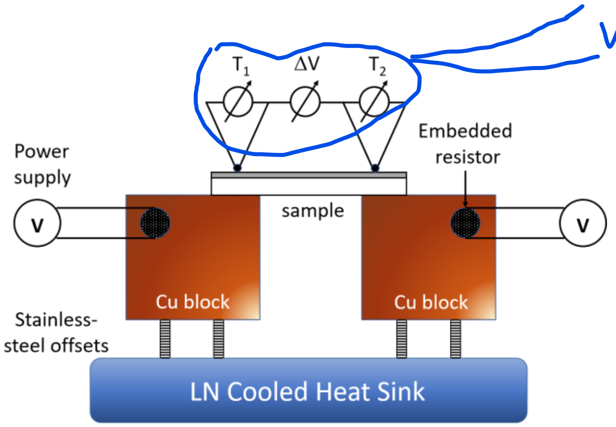

Figure S1. Schematic of Seebeck coefficient sample stage assembly.

Level selection and design scheme

CuZnS thin films have been made by different deposition methods in our previous work.[3] With chemical bath deposition, we particularly studied the influence of Cu content and Na_2EDTA concentration on film quality. In an initial screen to inform our DOE for the present work, we varied these parameters over a wide range as shown in Figure S2. Cu precursor contents Cu/(Cu+Zn) lower than 0.65 or greater than 0.85 produce films with poor FOMs. This is understandable, as the hole conduction is due to the CuS in the composite structure but a tradeoff exists because this phase is more strongly absorbing in the visible than ZnS. Also, Na_2EDTA concentrations >0.1 M produce low FOMs due to larger grain sizes. We therefore bounded the Cu content and Na2EDTA concentration in the shaded regions of Figure S2 in the first round of experiments.

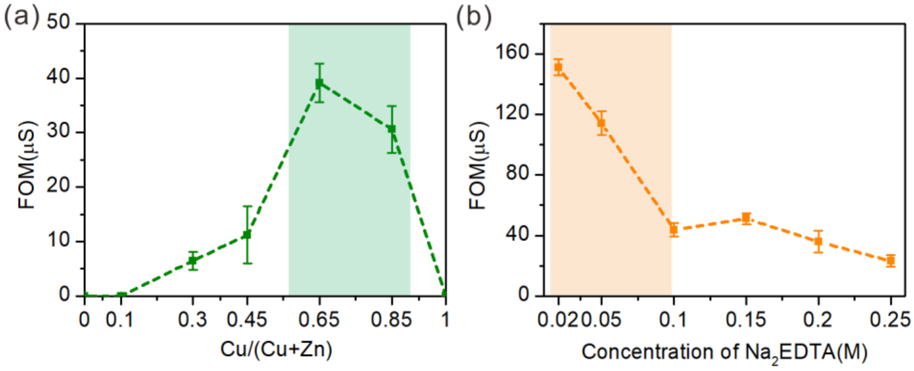

Figure S2. Relationships between Cu content (a) and Na_2EDTA concentration (b) with FOM. For (a), the Na_2EDTA concentration was 0.1 M and for (b) the Cu precursor content was with 0.65. The deposition temperature and time were 80 ºC and 60 minutes, respectively for all experiments. At least 6 samples were evaluated under each condition.

The balanced and orthogonal design for 4 levels each for reaction temperature, reaction time and concentration of Na_2EDTA is shown in Figure S3. Three axes represent each of the factors (reaction temperature, reaction time and Na2EDTA concentration). Levels are equally distributed to form a balanced subspace. At the same time, each factor is evaluated independently of all the other factors to achieve orthogonality.

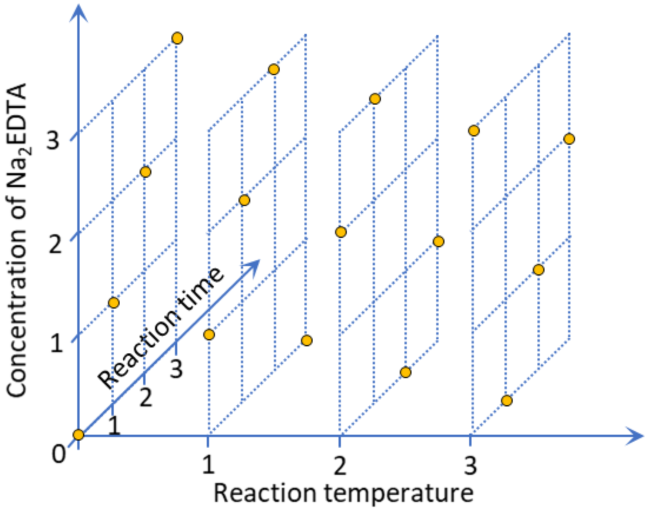

Figure S3. The first round of experiments design scheme for reaction temperature, reaction time and concentration of Na_2EDTA each with four levels. Cu/(Cu+Zn)=0.65 and 0.85 share the same design. Coordinates 0-3 corresponding to four levels from the lowest value to the highest value of each factor. The actual design sets are indicated by yellow balls.

Selected literature FOM values for p-TCMs

Table S1. FOMs of selected p-type transparent conducting materials computed using Eq. (1) in main text. (PLD: pulsed laser deposition; MBE: molecular beam epitaxy; CBD: chemical bath deposition).

Material	FOM (Φ_H, μS)	Deposition method	Reference
$CuCrO^{2+x}$	0.761	Combustion synthesis method	[4]
$Mg_xCr_{2-x}O_3$	0.493	MBE	[5]
$AgCrO_2$	0.009	CBD	[6]
$Cu_{0.4}CrO_2$	1.85	Spray pyrolysis	[7]
I-doped CuI	7.63	Sputtering	[8]
γ-CuI	6.42	Solid iodination method	[9]
$SrTiO_3$	7.78	PLD-MBE	[10]
Cu-Cr-O	0.002	Magnetron sputtering (annealing)	[11]
$Cu_xAl_{1-x}S_y$	174	CBD	[12]
NiO_x	0.004	RF sputtering	[13]
$Cu_xZn_{1-x}S$	138	PLD (annealing)	[14]
Cu alloyed ZnS	167	RF sputtering	[15]
Cu-Zn-S	216	CBD	The best in this work

ANOVA results from the first round of experiments

Table S2. ANOVA results from first round design of experiment.

Factors	DOF[a]	SSD[b]	F-ratio[c]	P-value[d]
Cu/(Cu+Zn)	1	48882	37.5	2.12e-8
Reaction temperature	3	27081	6.92	2.92e-4
Reaction time	3	27665	7.07	12.45e-4
Con. of Na_2EDTA	3	79745	20.4	2.95e-10
Residual	94	122600	NaN	NaN

[a] Degrees of freedom for each factor.

[b] Sum of squared deviations for each factor.

[c] F-ratio from F-test.

[d] P-value, probability of the results could have happened by chance.

Statistics and support vector regression (SVR) results from the second round of experiments

Table S3. Statistics results from second round design of experiment.

Factors	DOF[a]	SSD[b]	F-ratio[c]	P-value[d]
Cu/(Cu+Zn)	2	6277.3	7.00	1.55e-3
Reaction time	2	2712.0	3.02	5.40e-2
Reaction time	2	1960.5	2.19	1.19e-1
Con. of Na_2EDTA	2	64071	71.5	8.98e-19
Residual	83	37210	NaN	NaN

[a] Degrees of freedom for each factor.

[b] Sum of squared deviations for each factor.

[c] F-ratio from F-test.

[d] P-value, probability of the results could have happened by chance.

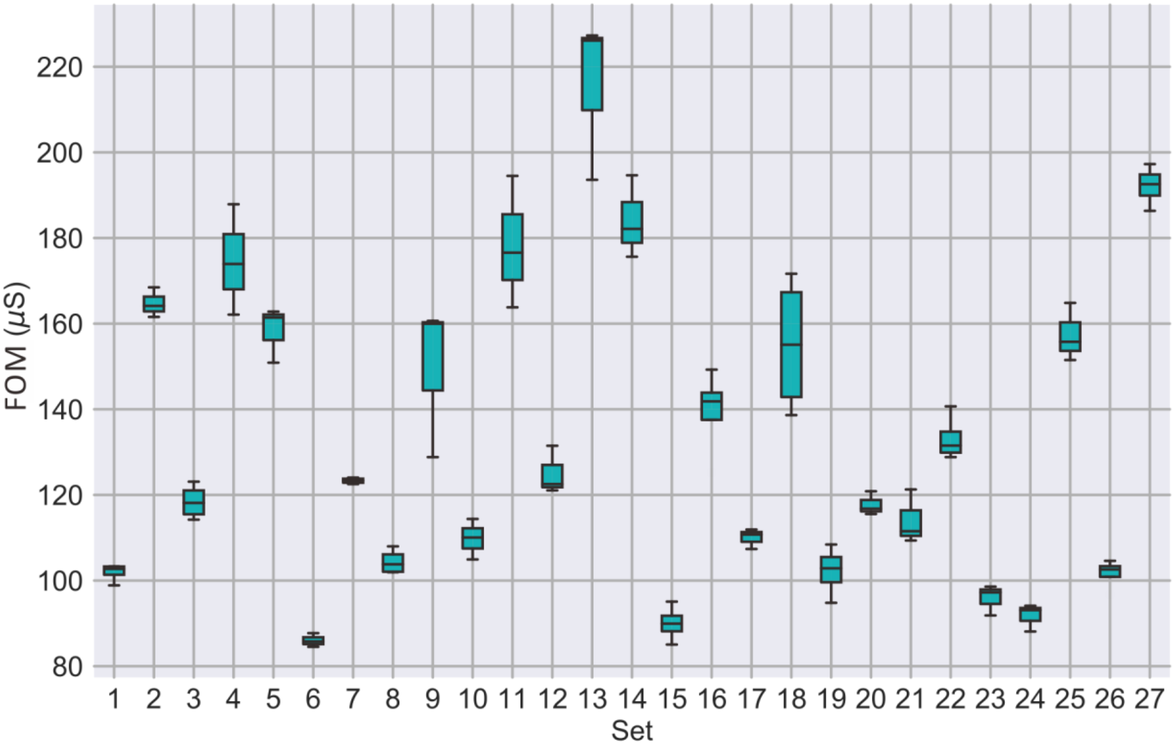

Figure S4. Boxplot of FOMs grouped by set from the second round DOE.

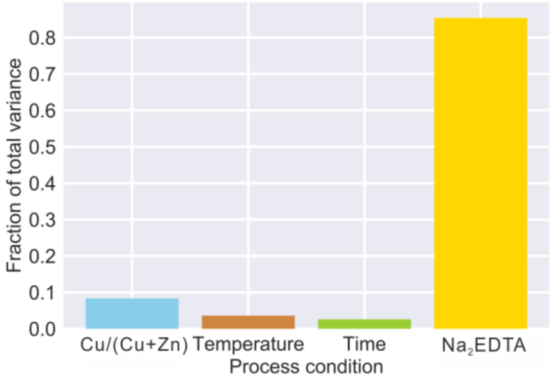

Figure S5. Analysis of variance and factor evaluation from the second round DOE.

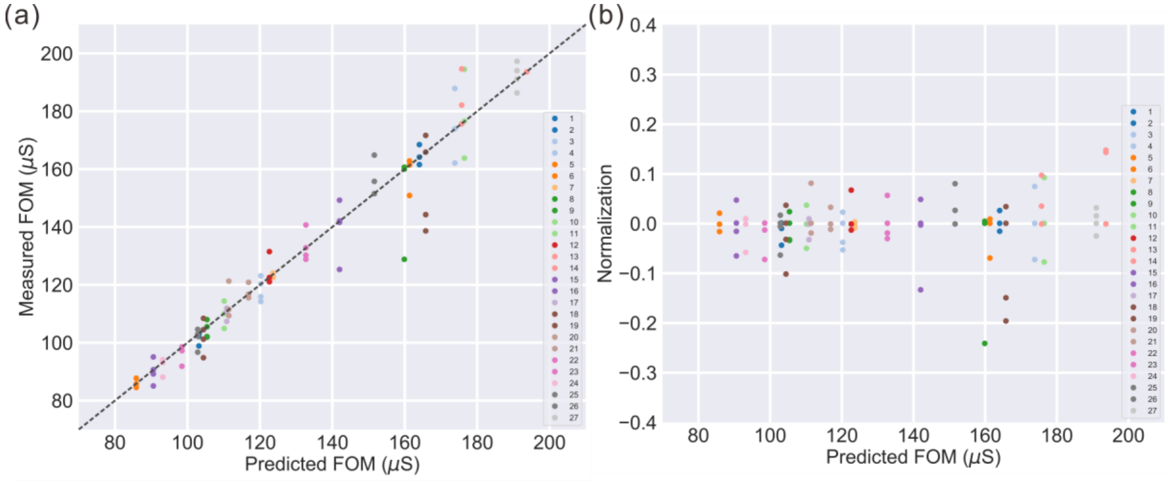

Figure S6. Support vector regression applied to the second round (a) Measured versus predicted FOM and (b) Normalized results versus predicted FOM.

Photovoltaic and photodiode device performance and statistics

For the heterojunction solar cells, we performed measurements on 15 1×1 cm^2 devices for each configuration (p-CZS/n-Si and p-CZS/p$^+$n-Si, optimized and non-optimized), for a total of 60 independent measurements. As shown in Figure S7(a), the optimized CZS films increase the J_{sc} by 20% and 45% for the p-CZS/n-Si and p-CZS/p$^+$n-Si geometries, respectively. Interestingly, for the p-CZS/n-Si structures, the V_{oc} with optimized p-CZS is ~20 mV lower (only ~4%), Figure S7(b). We also observe that the fill factors and efficiency for both structures are increased by optimized CZS films, Figure S7(c and d).

For p-CZS/n-ZnO photodiodes we tested 10 optimized and 10 non-optimized diodes. Figure S8 clearly shows that a higher rectification ratio (30 times) is obtained for devices with an optimized CZS film.

Statistical significance can be evaluated by an unpaired-sample t test.[16,17] Using heterojunction solar cell efficiency as an example, p-CZS/n-Si with optimized and non-optimized CZS films have different means of η_1=2.00 and η_2=1.62, and standard deviations of σ_1=0.058 and σ_2=0.055, respectively. The null hypothesis of no difference between means is rejected at the $\alpha = 0.05$ level as the calculated t statistic is 18.97 for 14 degrees of freedom, far larger than the critical value of 2.145. Similar statistical significance is found for the other solar cell parameters.

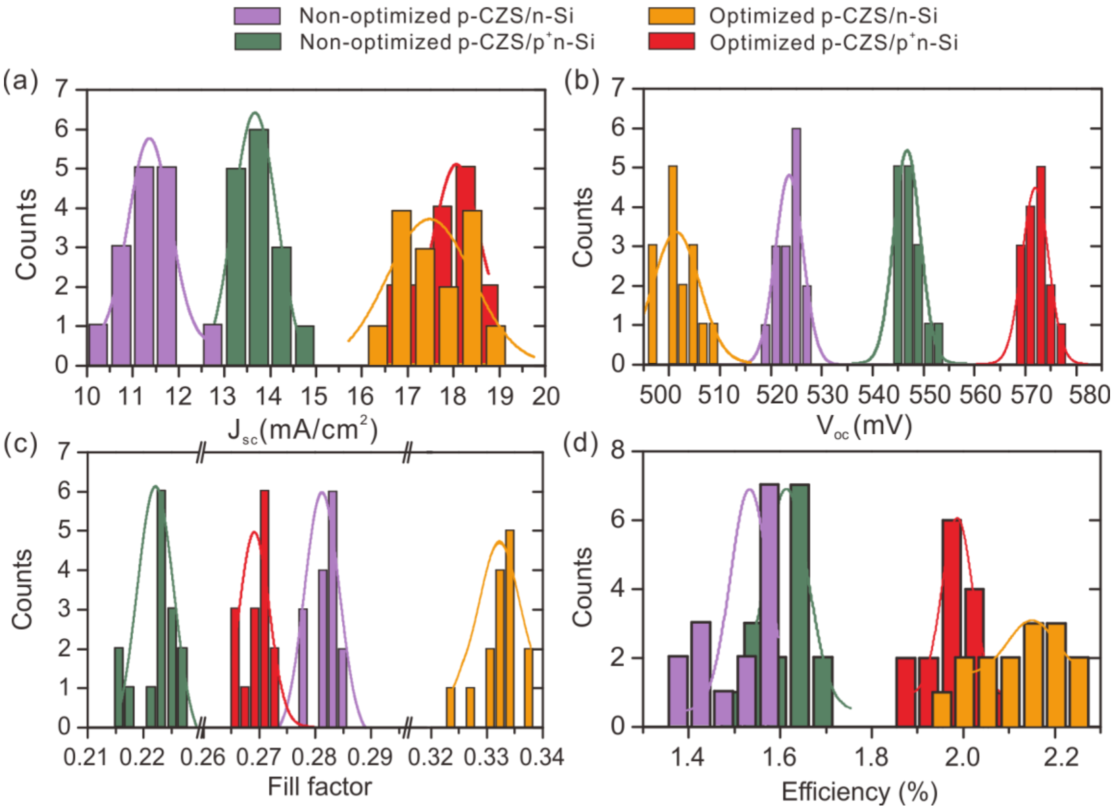

Figure S7. Histograms of the Gaussian distribution of (a) short circuit current, (b) open circuit voltage, (c) fill factor and (d) efficiency for p-CZS/n-Si and p-CZS/p$^+$n-Si solar cells. The total number of measured devices is 15 for each case. Fitting of normal distributions were done with Origin.

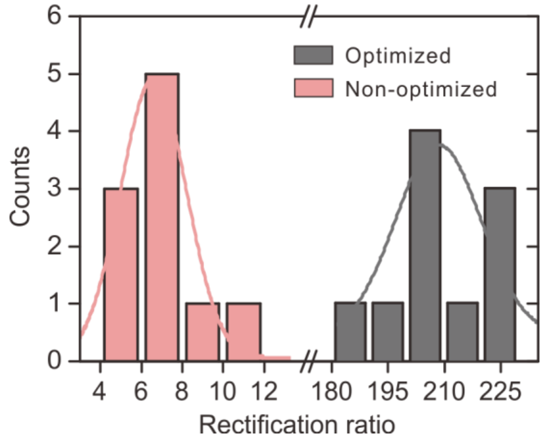

Figure S8. Histograms of the Gaussian distribution of p-CZS/n-ZnO photodiodes. The total number of measured devices is 10 for each case. Fitting was done with Origin.

Table S4. Statistics of solar cell performances. η is the mean and σ is the standard deviation.

Solar cell		Non-optimized p-CZS/n-Si	Optimized p-CZS/n-Si	Non-optimized p-CZS/p$^+$n-Si	Optimized p-CZS/p$^+$n-Si
J_{sc} (mA/cm^2)	η	11.3	17.5	13.7	18.0
	σ	0.618	0.673	0.464	0.686
V_{oc} (V)	η	523	506	547	572
	σ	2.42	3.57	2.33	2.52
FF	η	0.282	0.333	0.222	0.270
	σ	0.002	0.004	0.004	0.003
Efficiency (%)	η	1.62	2.00	1.54	2.15
	σ	0.055	0.058	0.079	0.095

Table S5. Statistics of p-CZS/n-ZnO photodiode performances. η is the mean value and σ is the standard deviation.

Photodiode		Non-optimized p-CZS/n-ZnO	Optimized p-CZS/n-ZnO
Rectification ratio	η	7.17	209
	σ	1.93	13.2

Discussion of support vector regression (SVR)[18–20]

Support vector machine (SVM) methods are used for classification but can also be used for regression. Support vector regression (SVR) maintains the key features of SVM such as individualizing the hyperplane, minimizing error and maximizing the margin. In SVM methods, part of the error is tolerated, namely that which lies within ε of the classification hyperplane or regression surface. A larger value of ε will lead to fewer of the data points being used as support vectors. For this reason, the choice of ε is important when considering the tradeoff between underfitting and overfitting. As shown in the cross-validation plot in Figure S4, use of large values of ε in for the data in this work lead to high loss, which is undesirable. For regression problem, errors in range y_i-$\omega^T\phi(x_i)$-b $\leq \varepsilon$ and $\omega^T\phi(x_i)$+b $\geq \varepsilon$ are not accounted for the total loss, these data points are thus not support vectors. When considering the total loss, only support vector matters, which means the SVR solves problem to get the following equation from support vectors.

$$\min_{\omega,b,\xi,\xi*} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

Statistics and SVR codes

```python
# basic packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# statistic module
import statsmodels.api as sm
from statsmodels.formula.api import ols

# The machine learning modules
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn import svm

df = pd.read_csv("experimental data.csv")   # read data

# this makes a box plot sorted by set and formats it
df.boxplot('FOM', by='Set', vert=True,
               boxprops = {'facecolor':'blue', 'color':'blue'},
               medianprops = {'linestyle':'-','color':'red'})
plt.style.use('seaborn')
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
font = {'size': 18}
plt.xlabel('Set', font)
plt.ylabel(r'FOM ($\mu$S)', font)
plt.title('1st_Run FOM')

model = ols('FOM ~ C(Cu_content) + C(Temperature) + C(Time) + C(EDTA)', df).fit()
# use ols = ordinary least squares
model.summary()
res = sm.stats.anova_lm(model, typ= 1)
print(res)   # this write the model summary to the console

# Now make the bar graph
objects = (df.columns[3],
           df.columns[4],
           df.columns[5],
           df.columns[6])
y_pos = np.arange(len(objects))
totalSSRnoRes = sum(res.sum_sq)-res.sum_sq[-1]
performance = [res.sum_sq[0]/totalSSRnoRes,
               res.sum_sq[1]/totalSSRnoRes,
               res.sum_sq[2]/totalSSRnoRes,
               res.sum_sq[3]/totalSSRnoRes]   # calculated bar lengths
plt.figure(2)
plt.bar(y_pos, performance,
        align='center',
        width=0.8,
        alpha=1.0,
        color=['skyblue', 'peru', 'yellowgreen', 'gold'])
plt.xticks(y_pos, objects)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
font = {'size': 18}
plt.xlabel('Process condition', font)
plt.ylabel('Fraction of total variance', font)
plt.title('1st run ANOVA results', font)
plt.style.use('seaborn')
plt.show()

names = ('Cu_content', 'Temperature','Time', 'EDTA')
variables = df.loc[:, names]
reg_FOM = Pipeline([('scl', StandardScaler()),
                    ('clf', svm.SVR(kernel='rbf', gamma=0.5,
                                    C=40, epsilon = 0.1,
                                    verbose=True))])
# Define a Pipeline that scales the data and applies the model
# C and gamma are determined via cross validation
reg_FOM.fit(variables, df.FOM)   # Fit the variables to the FOM
df['FOM_pred_svm'] = reg_FOM.predict(variables)
# Add the predicted FOM as a new column in df

colors = plt.cm.tab20(np.linspace(0, 1, 35)[0:len(df.Set.unique())])
color_dic = {label: color for label, color in zip(df.Set.unique(), colors)}
df['color'] = df.Set.map(color_dic)
# define a group of colors and associates a color with the labels in the dataframe.

# Now make the plot of predicted vs. measured FOM
fig, ax1 = plt.subplots(1, 1,
                        clear=True,
                        num='Predicted vs measured FOM',
                        figsize=(8, 6))
for label, data in df.groupby('Set'):
    plt.plot('FOM_pred_svm', 'FOM', 'o',
             color=data['color'].iloc[0],
             data=data,
             label=label)
# loop through to catch each experimental condition
plt.legend(loc='upper left', frameon=True)
plt.plot([-1, 220], [-1, 220], ls="--", c=".3")   # This draws a dotted line
plt.autoscale(enable=False)
plt.xlim(0,220)
plt.ylim(0,220)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
font = {'size': 18}
ax1.set_xlabel(r'Predicted FOM ($\mu$S)', font)
ax1.set_ylabel(r'Measured FOM ($\mu$S)', font)
plt.style.use('seaborn')
plt.tight_layout()
plt.show()

# Now make the plot of normalized measured-predicted vs. measured FOM
plt.figure(3)
fig, ax1 = plt.subplots(1, 1,
                        clear=True,
                        num='Predicted vs measured FOM',
                        figsize=(8, 6))
df['min'] = (df.FOM-df.FOM_pred_svm)/df.FOM
for label, data in df.groupby('Set'):
    plt.plot('FOM_pred_svm', 'min', 'o',
             color=data['color'].iloc[0],
             data=data,
             label=label)
plt.legend(loc='lower right', frameon=True)
plt.xlim(0,220)
plt.ylim(-0.4,0.4)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
font = {'size': 18}
ax1.set_ylabel('Normalization', font)
ax1.set_xlabel(r'Predicted FOM ($\mu$S)', font)
plt.style.use('seaborn')
plt.tight_layout()
plt.show()

# Plot slices of the 3D fit with value maps
u_len = 4
v_len = 2
vs = np.array([0.65,0.85]   # for 2 Cu contents
us = np.array([60,70,80,90]   # for 4 reaction temperatures
x_len, y_len = 100, 100   # number of points to make the contour plot
xs = np.linspace(10, 130, x_len)
ys = np.linspace(0.01, 0.11, y_len)
vi, ui, xi, yi = names
fig, axs = plt.subplots(nrows=1, ncols=u_len,
                        sharex=True, sharey=True,   # Same x and y axes
                        clear=True,
                        num='Support Vector Machine Regression, FOM',
                        figsize=(20, 4.5))
for ax, u in zip(axs, us):
    xm, ym = np.meshgrid(xs, ys)
    vm = 0.65 * np.ones_like(xm)   # Replace with each Cu content
    um = u * np.ones_like(xm)
    r = np.c_[vm.flatten(), um.flatten(), xm.flatten(), ym.flatten()]
    c = reg_FOM.predict(r).reshape(x_len, y_len)
# feed flattened mesh r to the predication algorithm
# then reshape the predictions back to a matrix
# Now make a contour map
    cmap = ax.contour(xs, ys, c, vmin=0, vmax=220, cmap='gray_r')
    plt.clabel(cmap, inline=1, fontsize=13)   # Make a value map
    pmap = ax.pcolormesh(xs, ys, c,
                         shading='gouraud',
                         vmin=0, vmax=220,
                         cmap='viridis')
    df.Time = pd.to_numeric(df.Time)
    for label, data in df.query('Temperature==@u and Cu_content==0.65').groupby('Set'):
        ax.plot('Time', 'EDTA', 'o',
                color=data['color'].iloc[0],
                data=data.iloc[0],
                mec='k',
                mew=0.5,
                label=label)
    ax.legend(loc='upper left', frameon=True)
    font = {'size': 18}
    ax.set_ylabel(f'{yi} (M)', font)
    ax.set_xlabel(f'{xi} (min)', font)
    plt.xticks(fontsize=18)
    plt.yticks(fontsize=18)
plt.tight_layout()
plt.colorbar(pmap, ax=axs, fraction=0.04)   # add color bar.
plt.show()
```

Cross validation and learning curve codes

```
# Basic packages
import numpy as np
import pandas as pd
from sklearn.learning_curve import validation_curve
from sklearn import preprocessing   # module used for normalization
import matplotlib.pyplot as plt

# The machine learning modules
from sklearn import svm

df = pd.read_csv("experimental data.csv")
data = df[['Cu_content','Temperature','Time','EDTA']]
target = df[['FOM']]
X = data
y = target

#test C value
param_range = np.linspace(1,101,50)   # set the range for parameter C
train_loss, test_loss = validation_curve(
        svm.SVR(kernel='rbf', gamma=0.5),
        preprocessing.scale(X),preprocessing.scale(y.values.ravel()),param_name='C',
        param_range=param_range, cv=10,
        scoring = 'neg_mean_squared_error')
train_loss_mean = -np.mean(train_loss, axis=1)
test_loss_mean = -np.mean(test_loss, axis=1)
# avoid negatives

# make the learning curve for C
plt.figure(1)
plt.plot(param_range, train_loss_mean, 'o-', color="r", label="Training")
plt.plot(param_range, test_loss_mean, 'o-', color="g", label="Cross_validation")
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
font = {'size':15}
plt.xlabel("C")
plt.ylabel("Loss")
plt.legend(loc="best",fontsize=15)
plt.show()

#test gamma value
param_range = np.linspace(0,1,50)   # set the range for parameter gamma
train_loss, test_loss = validation_curve(
        svm.SVR(kernel='rbf', C=40),
        preprocessing.scale(X), preprocessing.scale(y.values.ravel()),
        param_name='gamma',
        param_range=param_range, cv=10,
        scoring = 'neg_mean_squared_error')
train_loss_mean = -np.mean(train_loss, axis=1)
test_loss_mean = -np.mean(test_loss, axis=1)

# make the learning curve for gamma
plt.figure(2)
plt.plot(param_range, train_loss_mean, 'o-', color="r", label="Training")
plt.plot(param_range, test_loss_mean, 'o-', color="g", label="Cross_validation")
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
font = {'size':15}
plt.xlabel("gamma")
plt.ylabel("Loss")
plt.legend(loc="best",fontsize=15)
plt.show()

#test epsilon value
param_range = np.linspace(0,1,50)   # set the range for parameter gamma
train_loss, test_loss = validation_curve(
        svm.SVR(kernel='rbf', C=40, gamma=0.5)
        preprocessing.scale(X), preprocessing.scale(y.values.ravel()),
        param_name='epsilon',
        param_range=param_range, cv=10,
        scoring = 'neg_mean_squared_error')
train_loss_mean = -np.mean(train_loss, axis=1)
test_loss_mean = -np.mean(test_loss, axis=1)

# make the learning curve for gamma
plt.figure(3)
plt.plot(param_range, train_loss_mean, 'o-', color="r", label="Training")
plt.plot(param_range, test_loss_mean, 'o-', color="g", label="Cross_validation")
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
font = {'size':15}
plt.xlabel("epsilon")
plt.ylabel("Loss")
plt.legend(loc="best",fontsize=15)
plt.show()
```

References

(1) Brandt, M.; Herbst, P.; Angerer, H.; Ambacher, O.; Stutzmann, M. Thermopower Investigation of N- and p-Type GaN. *Phys. Rev. B - Condens. Matter Mater. Phys.* **1998**, *58* (12), 7786–7791. https://doi.org/10.1103/PhysRevB.58.7786.

(2) Miller, N. R. Thermopower of P-Type Indium Nitride, University of California, Berkeley, 2009.

(3) Xu, X.; Bullock, J.; Schelhas, L. T.; Stutz, E. Z.; Fonseca, J. J.; Hettick, M.; Pool, V. L.; Tai, K. F.; Toney, M. F.; Fang, X.; et al. Chemical Bath Deposition of P-Type Transparent, Highly Conducting (CuS)x:(ZnS)1-XNanocomposite Thin Films and Fabrication of Si Heterojunction Solar Cells. *Nano Lett.* **2016**. https://doi.org/10.1021/acs.nanolett.5b05124.

(4) Wang, J.; Daunis, T. B.; Cheng, L.; Zhang, B.; Kim, J.; Hsu, J. W. P. Combustion Synthesis of P-Type Transparent Conducting CuCrO 2+ x and Cu:CrO x Thin Films at 180 °C. *ACS Appl. Mater. Interfaces* **2018**, *10* (4), 3732–3738. https://doi.org/10.1021/acsami.7b13680.

(5) Norton, E.; Farrell, L.; Callaghan, S. D.; McGuinness, C.; Shvets, I. V.; Fleischer, K. X-Ray Spectroscopic Studies of the Electronic Structure of Chromium-Based <math> <mi>p</Mi> </Math> -Type Transparent Conducting Oxides. *Phys. Rev. B* **2016**, *93* (11), 115302. https://doi.org/10.1103/PhysRevB.93.115302.

(6) Wei, R.; Tang, X.; Hu, L.; Yang, J.; Zhu, X.; Song, W.; Dai, J.; Zhu, X.; Sun, Y. Facile Chemical Solution Synthesis of P-Type Delafossite Ag-Based Transparent Conducting AgCrO 2 Films in an Open Condition. *J. Mater. Chem. C* **2017**, *5* (8), 1885–1892. https://doi.org/10.1039/C6TC04848J.

(7) Norton, E.; Farrell, L.; Zhussupbekova, A.; Mullarkey, D.; Caffrey, D.; Papanastasiou, D. T.; Oser, D.; Bellet, D.; Shvets, I. V.; Fleischer, K. Bending Stability of Cu0.4CrO2 - A Transparent p-Type Conducting Oxide for Large Area Flexible Electronics. *AIP Adv.* **2018**. https://doi.org/10.1063/1.5027038.

(8) Yang, C.; Souchay, D.; Kneiß, M.; Bogner, M.; Wei, H. M.; Lorenz, M.; Oeckler, O.; Benstetter, G.; Fu, Y. Q.; Grundmann, M. Transparent Flexible Thermoelectric Material Based on Non-Toxic Earth-Abundant p-Type Copper Iodide Thin Film. *Nat. Commun.* **2017**. https://doi.org/10.1038/ncomms16076.

(9) Yamada, N.; Ino, R.; Ninomiya, Y. Truly Transparent P-Type γ-CuI Thin Films with High Hole Mobility. *Chem. Mater.* **2016**, *28* (14), 4971–4981. https://doi.org/10.1021/acs.chemmater.6b01358.

(10) Huang, W.; Nechache, R.; Li, S.; Chaker, M.; Rosei, F. Electrical and Optical Properties of Transparent Conducting p -Type SrTiO 3 Thin Films. *J. Am. Ceram. Soc.* **2016**, *99* (1), 226–233. https://doi.org/10.1111/jace.13949.

(11) Sun, C. H.; Tsai, D. C.; Chang, Z. C.; Chen, E. C.; Shieu, F. S. Structural, Optical, and Electrical Properties of Conducting p-Type Transparent Cu–Cr–O Thin Films. *Ceram. Int.* **2016**. https://doi.org/10.1016/j.ceramint.2016.05.168.

(12) Dai, X.; Lei, H.; Chen, C.; Guo, Y.; Fang, G. A Simple Synthesis of Transparent and Highly

Conducting P-Type Cu x Al 1−x S y Nanocomposite Thin Films as the Hole Transporting Layer for Organic Solar Cells. *RSC Adv.* **2018**, *8* (30), 16887–16896. https://doi.org/10.1039/C8RA01299G.

(13) Grilli, M. L.; Menchini, F.; Dikonimos, T.; Nunziante, P.; Pilloni, L.; Yilmaz, M.; Piegari, A.; Mittiga, A. Effect of Growth Parameters on the Properties of RF-Sputtered Highly Conductive and Transparent p-Type NiO x Films. *Semicond. Sci. Technol.* **2016**, *31* (5), 055016. https://doi.org/10.1088/0268-1242/31/5/055016.

(14) Feng, M.; Zhou, H.; Guo, W.; Zhang, D.; Ye, L.; Li, W.; Ma, J.; Wang, G.; Chen, S. Fabrication of P-Type Transparent Conducting CuxZn1-XS Films on Glass Substrates with High Conductivity and Optical Transparency. *J. Alloys Compd.* **2018**, *750*, 750–756. https://doi.org/10.1016/j.jallcom.2018.03.402.

(15) Maurya, S. K.; Liu, Y.; Xu, X.; Woods-Robinson, R.; Das, C.; Ager, J. W.; Balasubramaniam, K. R. High Figure-of-Merit p -Type Transparent Conductor, Cu Alloyed ZnS via Radio Frequency Magnetron Sputtering. *J. Phys. D. Appl. Phys.* **2017**, *50* (50), 505107. https://doi.org/10.1088/1361-6463/aa95b3.

(16) Luber, E. J.; Buriak, J. M. Reporting Performance in Organic Photovoltaic Devices. *ACS Nano* **2013**, *7* (6), 4708–4714. https://doi.org/10.1021/nn402883g.

(17) Smestad, G. P.; Krebs, F. C.; Lampert, C. M.; Granqvist, C. G.; Chopra, K. L.; Mathew, X.; Takakura, H. Reporting Solar Cell Efficiencies in Solar Energy Materials and Solar Cells. *Sol. Energy Mater. Sol. Cells* **2008**, *92* (4), 371–373. https://doi.org/10.1016/j.solmat.2008.01.003.

(18) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in {P}ython. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

(19) Smola, A. J.; Schölkopf, B. A Tutorial on Support Vector Regression. *Stat. Comput.* **2004**, *14* (3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

(20) Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning*; Cambridge University Press: Cambridge, 2014. https://doi.org/10.1017/CBO9781107298019.