In [ ]:

```
import os
import pandas as pd
import fnmatch
import numpy as np

root = "C:/Users/c21012241/Dropbox"

### 13 Features ###
path = root + "/13 Features - Binary Crystals/\
C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 13U LR 10^-3 E 200 MB 1024 - 12 of 12"

#path = root +"/13 Features - Continuous Crystals/\
#C2 LR 10^-3 Epochs 200 MiniBatch 1024 - Hybrid LR 10^-3 Epochs 50 MiniBatch 1024 - 12 of 12"

#path = root + "/13 Features - Continuous Crystals/\
#C2 LR 10^-3 E 150 MB 1024 - H LR 10^-3 E 15 MB 1024 - 13Un LR 10^-3 E 150 MB 1024 - 12of12"

#path = root +"/13 Features - Binary Crystals/\
#C2 LR10^-3 E200 MB1024 - H LR10^-3 E15 MB1024 - 13U LR10^-3 E200 MB1024 - 12 of 12"

### 12 Features ###
#path = root + "/12 Features - Binary Crystals + No Brightness/\
#C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 12U LR 10^-3 E 200 MB 1024 - 12 of 12"

#path = root + "/12 Features - Continuous Crystals + No Bright/\
#C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024 - 12U LR10^-3 E200 MB1024 - 12of12"
```

In [ ]:

```
rockNamesTen = ["Granite", "Obsidian", "Pegmatite", "Pumice", "Gneiss", "Marble", "Slate", "Breccia", "Conglomerate", "Sandstone"]
Val_Hybrid_1_2_3 = []
Val_Hybrid_4_5_6 = []
Val_Hybrid_7_8_9 = []
Val_Hybrid_10_11_12 = []
Val_Hybrid_1_4_7 = []
Val_Hybrid_5_8_10 = []
Val_Hybrid_2_9_11 = []
Val_Hybrid_3_6_12 = []
Val_Hybrid_1_6_9 = []
Val_Hybrid_2_7_10 = []
Val_Hybrid_3_8_11 = []
Val_Hybrid_4_5_12 = []
Val_1_2_3 = []
Val_4_5_6 = []
Val_7_8_9 = []
Val_10_11_12 = []
Val_1_4_7 = []
Val_5_8_10 = []
Val_2_9_11 = []
Val_3_6_12 = []
Val_1_6_9 = []
Val_2_7_10 = []
Val_3_8_11 = []
Val_4_5_12 = []
keyword_01_02_03 = "*Val_1 2 3*"
keyword_04_05_06 = "*Val_4 5 6*"
keyword_07_08_09 = "*Val_7 8 9*"
keyword_10_11_12 = "*Val_10 11 12*"
keyword_01_04_07 = "*Val_1 4 7*"
keyword_05_08_10 = "*Val_5 8 10*"
keyword_02_09_11 = "*Val_2 9 11*"
keyword_03_06_12 = "*Val_3 6 12*"
keyword_01_06_09 = "*Val_1 6 9*"
keyword_02_07_10 = "*Val_2 7 10*"
keyword_03_08_11 = "*Val_3 8 11*"
keyword_04_05_12 = "*Val_4 5 12*"
keywordConfusion = '*Confusion*_Matrix*'
all_Confusion = []
C2_Accuracy = []
hybrid_Accuracy = []
all_Confusion_DF = []
```

In [ ]:

```
#Get all confusion matrix adn append to all_Confusion
for root, dirs, files in os.walk(path):
    for filename in fnmatch.filter(files, keywordConfusion):
        file_path = os.path.join(root, filename)
        all_Confusion.append(file_path)

# Sort all by date
all_Confusion.sort(key=os.path.getmtime)

# Walk through the sorted list and if a validation set keyword matches then add it to the relevant list
for file in all_Confusion:
    if fnmatch.fnmatch(file, keyword_01_02_03):
        df = pd.read_csv(file, header=None)
        Val_1_2_3.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_06):
        df = pd.read_csv(file, header=None)
        Val_4_5_6.append(df)
    elif fnmatch.fnmatch(file, keyword_07_08_09):
        df = pd.read_csv(file, header=None)
        Val_7_8_9.append(df)
    elif fnmatch.fnmatch(file, keyword_10_11_12):
        df = pd.read_csv(file, header=None)
        Val_10_11_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_04_07):
        df = pd.read_csv(file, header=None)
        Val_1_4_7.append(df)
    elif fnmatch.fnmatch(file, keyword_05_08_10):
        df = pd.read_csv(file, header=None)
        Val_5_8_10.append(df)
    elif fnmatch.fnmatch(file, keyword_02_09_11):
        df = pd.read_csv(file, header=None)
        Val_2_9_11.append(df)
    elif fnmatch.fnmatch(file, keyword_03_06_12):
        df = pd.read_csv(file, header=None)
        Val_3_6_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_06_09):
        df = pd.read_csv(file, header=None)
        Val_1_6_9.append(df)
    elif fnmatch.fnmatch(file, keyword_02_07_10):
        df = pd.read_csv(file, header=None)
        Val_2_7_10.append(df)
    elif fnmatch.fnmatch(file, keyword_03_08_11):
        df = pd.read_csv(file, header=None)
        Val_3_8_11.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_12):
        df = pd.read_csv(file, header=None)
        Val_4_5_12.append(df)

for file in all_Confusion:
    if fnmatch.fnmatch(file, keywordConfusion):
        df = pd.read_csv(file, header=None)
        all_Confusion_DF.append(df)
```

In [ ]:

```
def sumConfusionDfC2(Confusion, all_Confusion_DF):
    for df in all_Confusion_DF:
        df = df.apply(pd.to_numeric, errors='coerce')
        a = df.iloc[2:12, 0:10]
        Confusion = Confusion.add(a, fill_value=0)
    return Confusion

def sumConfusionDfHybrid(Confusion, all_Confusion_DF):
    for df in all_Confusion_DF:
        df = df.apply(pd.to_numeric, errors='coerce')
        a = df.iloc[2:12, 10:20]
        Confusion = Confusion.add(a, fill_value=0)
    return Confusion

def sumConfusionDfUnconstrained(Confusion, all_Confusion_DF):
    for df in all_Confusion_DF:
        df = df.apply(pd.to_numeric, errors='coerce')
        a = df.iloc[2:12, 20:30]
        Confusion = Confusion.add(a, fill_value=0)
    return Confusion
```

In [ ]:

```
C2Confusion = pd.DataFrame()
C2Confusion = sumConfusionDfC2(C2Confusion, all_Confusion_DF)
C2Confusion = C2Confusion.reset_index(drop=True)
C2Confusion.index = rockNamesTen
C2Confusion.columns = rockNamesTen
file_path = path + "/C2Confusion" + ".csv"

if os.path.isfile(file_path):
    # If the file already exists, create a new one with "_1" appended
    root, ext = os.path.splitext(file_path)
    new_file_path = root + "_1" + ext
else:
    # If not, use the original file path
    new_file_path = file_path

C2Confusion.to_csv(new_file_path)
```

In [ ]:

```
import plotly.graph_objects as go

total_sum_C2_Confusion = np.sum(C2Confusion.values)
C2ConfusionPercentages = np.round(((C2Confusion.values/total_sum_C2_Confusion) * 1000),2)

z = C2ConfusionPercentages
x = rockNamesTen
y = rockNamesTen
z_text = [[str(y) for y in x] for x in z]

layout = {
        "title": "C2 Confusion Matrix - 13 Features - Binary Crystals - C2 LR10^-3 E200 MB1024 - 12 of 12",
        "xaxis": {"title": "Predicted value"},
        "yaxis": {"title": "Real value"}

}

fig = go.Figure(data=go.Heatmap(z=z, x=x, y=y, autocolorscale = False,
                                colorscale = [[0, 'rgb(255,255,255)'], [1, 'rgb(100,149,237)']],
                                hoverongaps = False), layout=layout)

# Add annotations
for i in range(len(y)):
    for j in range(len(x)):
        fig.add_annotation(
            text=str(z_text[i][j] + "%"),
            x=x[j],
            y=y[i],
            showarrow=False,
            font=dict(size=12),
            visible=True,
            xanchor='center',
            yanchor='middle'
        )

fig.show()
fig.write_image(path + "/" + "C2 Confusion Matrix.png")
```

In [ ]:

```
HybridConfusion = pd.DataFrame()
HybridConfusion = sumConfusionDfHybrid(HybridConfusion, all_Confusion_DF)
HybridConfusion = HybridConfusion.reset_index(drop=True)
HybridConfusion.index = rockNamesTen
HybridConfusion.columns = rockNamesTen
file_path = path + "/HybridConfusion" + ".csv"

if os.path.isfile(file_path):
    # If the file already exists, create a new one with "_1" appended
    root, ext = os.path.splitext(file_path)
    new_file_path = root + "_1" + ext
else:
    # If not, use the original file path
    new_file_path = file_path

HybridConfusion.to_csv(new_file_path)
```

In [ ]:

```
import plotly.graph_objects as go

total_sum_Hybrid_Confusion = np.sum(HybridConfusion.values)
HybridConfusionPercentages = np.round(((HybridConfusion.values/total_sum_Hybrid_Confusion) * 1000),2)

z = HybridConfusionPercentages
x = rockNamesTen
y = rockNamesTen
z_text = [[str(y) for y in x] for x in z]

layout = {
        "title": "Hybrid Confusion Matrix - 13 Features - Binary Crystals - C2 LR10^-3 E200 MB1025 - H LR10^-3 E15 MB1024 - 12 of 12",
        "xaxis": {"title": "Predicted value"},
        "yaxis": {"title": "Real value"}
}

fig = go.Figure(data=go.Heatmap(z=z, x=x, y=y, autocolorscale = False,
                                colorscale = [[0, 'rgb(255,255,255)'], [1, 'rgb(100,149,237)']],
                                hoverongaps = False), layout=layout)

# Add annotations
for i in range(len(y)):
    for j in range(len(x)):
        fig.add_annotation(
            text=str(z_text[i][j] + "%"),
            x=x[j],
            y=y[i],
            showarrow=False,
            font=dict(size=12),
            visible=True,
            xanchor='center',
            yanchor='middle'
        )

fig.show()
fig.write_image(path + "/" + "Hybrid Confusion Matrix.png")
```

In [ ]:
```
UnconstrainedConfusion = pd.DataFrame()
UnconstrainedConfusion = sumConfusionDfUnconstrained(UnconstrainedConfusion, all_Confusion_DF)
UnconstrainedConfusion = UnconstrainedConfusion.reset_index(drop=True)
UnconstrainedConfusion.index = rockNamesTen
UnconstrainedConfusion.columns = rockNamesTen
file_path = path + "/UnconstrainedConfusion" + ".csv"

if os.path.isfile(file_path):
    # If the file already exists, create a new one with "_1" appended
    root, ext = os.path.splitext(file_path)
    new_file_path = root + "_1" + ext
else:
    # If not, use the original file path
    new_file_path = file_path

UnconstrainedConfusion.to_csv(new_file_path)
```

In [ ]:
```
import plotly.graph_objects as go

total_sum_Hybrid_Confusion = np.sum(UnconstrainedConfusion.values)
HybridConfusionPercentages = np.round(((HybridConfusion.values/total_sum_Hybrid_Confusion) * 1000),2)

z = HybridConfusionPercentages
x = rockNamesTen
y = rockNamesTen
z_text = [[str(y) for y in x] for x in z]

layout = {
        "title": "Hybrid Confusion Matrix - 13 Features - Binary Crystals - C2 LR10^-3 E200 MB102512 Runs of 12 Alternating Validation Images - 12 of 12",
        "xaxis": {"title": "Predicted value"},
        "yaxis": {"title": "Real value"}

}

fig = go.Figure(data=go.Heatmap(z=z, x=x, y=y, autocolorscale = False,
                                colorscale = [[0, 'rgb(255,255,255)'], [1, 'rgb(100,149,237)']],
                                hoverongaps = False), layout=layout)

# Add annotations
for i in range(len(y)):
    for j in range(len(x)):
        fig.add_annotation(
            text=str(z_text[i][j] + "%"),
            x=x[j],
            y=y[i],
            showarrow=False,
            font=dict(size=12),
            visible=True,
            xanchor='center',
            yanchor='middle'
        )

fig.show()
fig.write_image(path + "/" + "Hybrid Confusion Matrix.png")
```

In [ ]:
```
# Function to split hybrid and C2 networks accuracies in
def splitAccuraciesToDfC2(constrainedC2ValAcc, validationSet):
    for df in validationSet:
        a = df.iloc[12,1:2]
        constrainedC2ValAcc = pd.concat([constrainedC2ValAcc, a], axis=0,ignore_index=True)
    return constrainedC2ValAcc

def splitAccuraciesToDfHybrid(hybridNetworkValAcc, validationSet):
    for df in validationSet:
        a = df.iloc[12,11:12]
        hybridNetworkValAcc = pd.concat([hybridNetworkValAcc, a], axis=0,ignore_index=True)
    return hybridNetworkValAcc
```

In [ ]:
```
Val_Acc_C2_1_2_3 = pd.DataFrame()
Val_Acc_C2_1_2_3 = splitAccuraciesToDfC2(Val_Acc_C2_1_2_3, Val_1_2_3)
Val_Acc_C2_4_5_6 = pd.DataFrame()
Val_Acc_C2_4_5_6 = splitAccuraciesToDfC2(Val_Acc_C2_4_5_6, Val_4_5_6)
Val_Acc_C2_7_8_9 = pd.DataFrame()
Val_Acc_C2_7_8_9 = splitAccuraciesToDfC2(Val_Acc_C2_7_8_9, Val_7_8_9)
Val_Acc_C2_10_11_12 = pd.DataFrame()
Val_Acc_C2_10_11_12 = splitAccuraciesToDfC2(Val_Acc_C2_10_11_12, Val_10_11_12)
Val_Acc_C2_1_4_7 = pd.DataFrame()
Val_Acc_C2_1_4_7 = splitAccuraciesToDfC2(Val_Acc_C2_1_4_7, Val_1_4_7)
Val_Acc_C2_5_8_10 = pd.DataFrame()
Val_Acc_C2_5_8_10 = splitAccuraciesToDfC2(Val_Acc_C2_5_8_10, Val_5_8_10)
Val_Acc_C2_2_9_11 = pd.DataFrame()
Val_Acc_C2_2_9_11 = splitAccuraciesToDfC2(Val_Acc_C2_2_9_11, Val_2_9_11)
Val_Acc_C2_3_6_12 = pd.DataFrame()
Val_Acc_C2_3_6_12 = splitAccuraciesToDfC2(Val_Acc_C2_3_6_12, Val_3_6_12)
Val_Acc_C2_1_6_9 = pd.DataFrame()
Val_Acc_C2_1_6_9 = splitAccuraciesToDfC2(Val_Acc_C2_1_6_9, Val_1_6_9)
Val_Acc_C2_2_7_10 = pd.DataFrame()
Val_Acc_C2_2_7_10 = splitAccuraciesToDfC2(Val_Acc_C2_2_7_10, Val_2_7_10)
Val_Acc_C2_3_8_11 = pd.DataFrame()
Val_Acc_C2_3_8_11 = splitAccuraciesToDfC2(Val_Acc_C2_3_8_11, Val_3_8_11)
Val_Acc_C2_4_5_12 = pd.DataFrame()
Val_Acc_C2_4_5_12 = splitAccuraciesToDfC2(Val_Acc_C2_4_5_12, Val_4_5_12)

Val_Acc_Hybrid_1_2_3 = pd.DataFrame()
Val_Acc_Hybrid_1_2_3 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_1_2_3, Val_1_2_3)
Val_Acc_Hybrid_4_5_6 = pd.DataFrame()
Val_Acc_Hybrid_4_5_6 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_4_5_6, Val_4_5_6)
Val_Acc_Hybrid_7_8_9 = pd.DataFrame()
Val_Acc_Hybrid_7_8_9 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_7_8_9, Val_7_8_9)
Val_Acc_Hybrid_10_11_12 = pd.DataFrame()
Val_Acc_Hybrid_10_11_12 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_10_11_12, Val_10_11_12)
Val_Acc_Hybrid_1_4_7 = pd.DataFrame()
Val_Acc_Hybrid_1_4_7 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_1_4_7, Val_1_4_7)
Val_Acc_Hybrid_5_8_10 = pd.DataFrame()
Val_Acc_Hybrid_5_8_10 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_5_8_10, Val_5_8_10)
Val_Acc_Hybrid_2_9_11 = pd.DataFrame()
Val_Acc_Hybrid_2_9_11 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_2_9_11, Val_2_9_11)
Val_Acc_Hybrid_3_6_12 = pd.DataFrame()
Val_Acc_Hybrid_3_6_12 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_3_6_12, Val_3_6_12)
Val_Acc_Hybrid_1_6_9 = pd.DataFrame()
Val_Acc_Hybrid_1_6_9 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_1_6_9, Val_1_6_9)
Val_Acc_Hybrid_2_7_10 = pd.DataFrame()
Val_Acc_Hybrid_2_7_10 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_2_7_10, Val_2_7_10)
Val_Acc_Hybrid_3_8_11 = pd.DataFrame()
Val_Acc_Hybrid_3_8_11 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_3_8_11, Val_3_8_11)
Val_Acc_Hybrid_4_5_12 = pd.DataFrame()
Val_Acc_Hybrid_4_5_12 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_4_5_12, Val_4_5_12)
```

In [ ]:
```
def meanValStdSem(valSet):
    valSet_means = np.mean((valSet.sum(axis=1)).to_numpy())
    valSet_std = (valSet.sum(axis=1)).to_numpy().std()
    valSet_sem = valSet_std / np.sqrt(np.size(valSet))

    return valSet_means, valSet_std, valSet_sem
```

In [ ]:
```
# Validation sets mean, standard deviation and standard error of the mean

totalVal_Acc_C2_1_2_3_mean_std_sem = meanValStdSem(Val_Acc_C2_1_2_3)
totalVal_Acc_C2_4_5_6_mean_std_sem = meanValStdSem(Val_Acc_C2_4_5_6)
totalVal_Acc_C2_7_8_9_mean_std_sem = meanValStdSem(Val_Acc_C2_7_8_9)
totalVal_Acc_C2_10_11_12_mean_std_sem = meanValStdSem(Val_Acc_C2_10_11_12)
totalVal_Acc_C2_1_4_7_mean_std_sem = meanValStdSem(Val_Acc_C2_1_4_7)
totalVal_Acc_C2_5_8_10_mean_std_sem = meanValStdSem(Val_Acc_C2_5_8_10)
totalVal_Acc_C2_2_9_11_mean_std_sem = meanValStdSem(Val_Acc_C2_2_9_11)
totalVal_Acc_C2_3_6_12_mean_std_sem = meanValStdSem(Val_Acc_C2_3_6_12)
totalVal_Acc_C2_1_6_9_mean_std_sem = meanValStdSem(Val_Acc_C2_1_6_9)
totalVal_Acc_C2_2_7_10_mean_std_sem = meanValStdSem(Val_Acc_C2_2_7_10)
totalVal_Acc_C2_3_8_11_mean_std_sem = meanValStdSem(Val_Acc_C2_3_8_11)
totalVal_Acc_C2_4_5_12_mean_std_sem = meanValStdSem(Val_Acc_C2_4_5_12)
```

In [ ]:
```
totalVal_Acc_hybrid_1_2_3_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_1_2_3)
totalVal_Acc_hybrid_4_5_6_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_4_5_6)
totalVal_Acc_hybrid_7_8_9_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_7_8_9)
totalVal_Acc_hybrid_10_11_12_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_10_11_12)
totalVal_Acc_hybrid_1_4_7_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_1_4_7)
totalVal_Acc_hybrid_5_8_10_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_5_8_10)
totalVal_Acc_hybrid_2_9_11_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_2_9_11)
totalVal_Acc_hybrid_3_6_12_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_3_6_12)
totalVal_Acc_hybrid_1_6_9_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_1_6_9)
totalVal_Acc_hybrid_2_7_10_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_2_7_10)
totalVal_Acc_hybrid_3_8_11_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_3_8_11)
totalVal_Acc_hybrid_4_5_12_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_4_5_12)
```

In [ ]:
```
meanOfMeansC2 = (totalVal_Acc_C2_1_2_3_mean_std_sem [0] + totalVal_Acc_C2_4_5_6_mean_std_sem [0] +
                 totalVal_Acc_C2_7_8_9_mean_std_sem [0] + totalVal_Acc_C2_10_11_12_mean_std_sem [0] +
                 totalVal_Acc_C2_1_4_7_mean_std_sem [0] + totalVal_Acc_C2_5_8_10_mean_std_sem [0] +
                 totalVal_Acc_C2_2_9_11_mean_std_sem [0] + totalVal_Acc_C2_3_6_12_mean_std_sem [0] +
                 totalVal_Acc_C2_1_6_9_mean_std_sem [0] + totalVal_Acc_C2_2_7_10_mean_std_sem [0] +
                 totalVal_Acc_C2_3_8_11_mean_std_sem [0] + totalVal_Acc_C2_4_5_12_mean_std_sem [0])/12

stdOfMeansC2 = (totalVal_Acc_C2_1_2_3_mean_std_sem [1] + totalVal_Acc_C2_4_5_6_mean_std_sem [1] +
                totalVal_Acc_C2_7_8_9_mean_std_sem [1] + totalVal_Acc_C2_10_11_12_mean_std_sem [1] +
                totalVal_Acc_C2_1_4_7_mean_std_sem [1] + totalVal_Acc_C2_5_8_10_mean_std_sem [1] +
                totalVal_Acc_C2_2_9_11_mean_std_sem [1] + totalVal_Acc_C2_3_6_12_mean_std_sem [1] +
                totalVal_Acc_C2_1_6_9_mean_std_sem [1] + totalVal_Acc_C2_2_7_10_mean_std_sem [1] +
                totalVal_Acc_C2_3_8_11_mean_std_sem [1] + totalVal_Acc_C2_4_5_12_mean_std_sem [1])/12

semOfMeansC2 = (totalVal_Acc_C2_1_2_3_mean_std_sem [2] + totalVal_Acc_C2_4_5_6_mean_std_sem [2] +
                totalVal_Acc_C2_7_8_9_mean_std_sem [2] + totalVal_Acc_C2_10_11_12_mean_std_sem [2] +
                totalVal_Acc_C2_1_4_7_mean_std_sem [2] + totalVal_Acc_C2_5_8_10_mean_std_sem [2] +
                totalVal_Acc_C2_2_9_11_mean_std_sem [2] + totalVal_Acc_C2_3_6_12_mean_std_sem [2] +
                totalVal_Acc_C2_1_6_9_mean_std_sem [2] + totalVal_Acc_C2_2_7_10_mean_std_sem [2] +
                totalVal_Acc_C2_3_8_11_mean_std_sem [2] + totalVal_Acc_C2_4_5_12_mean_std_sem [2])/12

meanOfMeansC2_12 = [totalVal_Acc_C2_1_2_3_mean_std_sem [0] , totalVal_Acc_C2_4_5_6_mean_std_sem [0] ,
                    totalVal_Acc_C2_7_8_9_mean_std_sem [0] , totalVal_Acc_C2_10_11_12_mean_std_sem [0] ,
                    totalVal_Acc_C2_1_4_7_mean_std_sem [0] , totalVal_Acc_C2_5_8_10_mean_std_sem [0] ,
                    totalVal_Acc_C2_2_9_11_mean_std_sem [0] , totalVal_Acc_C2_3_6_12_mean_std_sem [0] ,
                    totalVal_Acc_C2_1_6_9_mean_std_sem [0] , totalVal_Acc_C2_2_7_10_mean_std_sem [0] ,
                    totalVal_Acc_C2_3_8_11_mean_std_sem [0] , totalVal_Acc_C2_4_5_12_mean_std_sem [0]
                   ]

stdOfMeansC2_12 = [totalVal_Acc_C2_1_2_3_mean_std_sem [1] , totalVal_Acc_C2_4_5_6_mean_std_sem [1] ,
                   totalVal_Acc_C2_7_8_9_mean_std_sem [1] , totalVal_Acc_C2_10_11_12_mean_std_sem [1] ,
                   totalVal_Acc_C2_1_4_7_mean_std_sem [1] , totalVal_Acc_C2_5_8_10_mean_std_sem [1] ,
                   totalVal_Acc_C2_2_9_11_mean_std_sem [1] , totalVal_Acc_C2_3_6_12_mean_std_sem [1] ,
                   totalVal_Acc_C2_1_6_9_mean_std_sem [1] , totalVal_Acc_C2_2_7_10_mean_std_sem [1] ,
                   totalVal_Acc_C2_3_8_11_mean_std_sem [1] , totalVal_Acc_C2_4_5_12_mean_std_sem [1]
                  ]

semOfMeansC2_12 = [totalVal_Acc_C2_1_2_3_mean_std_sem [2] , totalVal_Acc_C2_4_5_6_mean_std_sem [2] ,
                   totalVal_Acc_C2_7_8_9_mean_std_sem [2] , totalVal_Acc_C2_10_11_12_mean_std_sem [2] ,
                   totalVal_Acc_C2_1_4_7_mean_std_sem [2] , totalVal_Acc_C2_5_8_10_mean_std_sem [2] ,
                   totalVal_Acc_C2_2_9_11_mean_std_sem [2] , totalVal_Acc_C2_3_6_12_mean_std_sem [2] ,
                   totalVal_Acc_C2_1_6_9_mean_std_sem [2] , totalVal_Acc_C2_2_7_10_mean_std_sem [2] ,
                   totalVal_Acc_C2_3_8_11_mean_std_sem [2] , totalVal_Acc_C2_4_5_12_mean_std_sem [2]
                  ]
```

In [ ]:
```
print("meanOfMeansC2 = " + str(meanOfMeansC2))
print("semOfMeansC2 = " + str(semOfMeansC2))
```

In [ ]:
```
meanOfMeansHybrid = (totalVal_Acc_hybrid_1_2_3_mean_std_sem [0] + totalVal_Acc_hybrid_4_5_6_mean_std_sem [0] +
                     totalVal_Acc_hybrid_7_8_9_mean_std_sem [0] + totalVal_Acc_hybrid_10_11_12_mean_std_sem [0] +
                     totalVal_Acc_hybrid_1_4_7_mean_std_sem [0] + totalVal_Acc_hybrid_5_8_10_mean_std_sem [0] +
                     totalVal_Acc_hybrid_2_9_11_mean_std_sem [0] + totalVal_Acc_hybrid_3_6_12_mean_std_sem [0] +
                     totalVal_Acc_hybrid_1_6_9_mean_std_sem [0] + totalVal_Acc_hybrid_2_7_10_mean_std_sem [0] +
                     totalVal_Acc_hybrid_3_8_11_mean_std_sem [0] + totalVal_Acc_hybrid_4_5_12_mean_std_sem [0])/12

stdOfMeansHybrid = (totalVal_Acc_hybrid_1_2_3_mean_std_sem [1] + totalVal_Acc_hybrid_4_5_6_mean_std_sem [1] +
                    totalVal_Acc_hybrid_7_8_9_mean_std_sem [1] + totalVal_Acc_hybrid_10_11_12_mean_std_sem [1] +
                    totalVal_Acc_hybrid_1_4_7_mean_std_sem [1] + totalVal_Acc_hybrid_5_8_10_mean_std_sem [1] +
                    totalVal_Acc_hybrid_2_9_11_mean_std_sem [1] + totalVal_Acc_hybrid_3_6_12_mean_std_sem [1] +
                    totalVal_Acc_hybrid_1_6_9_mean_std_sem [1] + totalVal_Acc_hybrid_2_7_10_mean_std_sem [1] +
                    totalVal_Acc_hybrid_3_8_11_mean_std_sem [1] + totalVal_Acc_hybrid_4_5_12_mean_std_sem [1])/12

semOfMeansHybrid = (totalVal_Acc_hybrid_1_2_3_mean_std_sem [2] + totalVal_Acc_hybrid_4_5_6_mean_std_sem [2] +
                    totalVal_Acc_hybrid_7_8_9_mean_std_sem [2] + totalVal_Acc_hybrid_10_11_12_mean_std_sem [2] +
                    totalVal_Acc_hybrid_1_4_7_mean_std_sem [2] + totalVal_Acc_hybrid_5_8_10_mean_std_sem [2] +
                    totalVal_Acc_hybrid_2_9_11_mean_std_sem [2] + totalVal_Acc_hybrid_3_6_12_mean_std_sem [2] +
                    totalVal_Acc_hybrid_1_6_9_mean_std_sem [2] + totalVal_Acc_hybrid_2_7_10_mean_std_sem [2] +
                    totalVal_Acc_hybrid_3_8_11_mean_std_sem [2] + totalVal_Acc_hybrid_4_5_12_mean_std_sem [2])/12

meanOfMeansHybrid_12 = [totalVal_Acc_hybrid_1_2_3_mean_std_sem [0] , totalVal_Acc_hybrid_4_5_6_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_7_8_9_mean_std_sem [0] , totalVal_Acc_hybrid_10_11_12_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_1_4_7_mean_std_sem [0] , totalVal_Acc_hybrid_5_8_10_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_2_9_11_mean_std_sem [0] , totalVal_Acc_hybrid_3_6_12_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_1_6_9_mean_std_sem [0] , totalVal_Acc_hybrid_2_7_10_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_3_8_11_mean_std_sem [0] , totalVal_Acc_hybrid_4_5_12_mean_std_sem [0]
                       ]

stdOfMeansHybrid_12 = [totalVal_Acc_hybrid_1_2_3_mean_std_sem [1] , totalVal_Acc_hybrid_4_5_6_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_7_8_9_mean_std_sem [1] , totalVal_Acc_hybrid_10_11_12_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_1_4_7_mean_std_sem [1] , totalVal_Acc_hybrid_5_8_10_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_2_9_11_mean_std_sem [1] , totalVal_Acc_hybrid_3_6_12_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_1_6_9_mean_std_sem [1] , totalVal_Acc_hybrid_2_7_10_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_3_8_11_mean_std_sem [1] , totalVal_Acc_hybrid_4_5_12_mean_std_sem [1]
                      ]

semOfMeansHybrid_12 = [totalVal_Acc_hybrid_1_2_3_mean_std_sem [2] , totalVal_Acc_hybrid_4_5_6_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_7_8_9_mean_std_sem [2] , totalVal_Acc_hybrid_10_11_12_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_1_4_7_mean_std_sem [2] , totalVal_Acc_hybrid_5_8_10_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_2_9_11_mean_std_sem [2] , totalVal_Acc_hybrid_3_6_12_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_1_6_9_mean_std_sem [2] , totalVal_Acc_hybrid_2_7_10_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_3_8_11_mean_std_sem [2] , totalVal_Acc_hybrid_4_5_12_mean_std_sem [2]
                      ]
```

In [ ]:
```
print("meanOfMeansC2 = " + str(meanOfMeansHybrid))
print("semOfMeansC2 = " + str(semOfMeansHybrid))
```

In [ ]:
```
print("Val 3, 6, 12 " + " C2 mean = " + str(meanOfMeansC2_12[7]) + " C2 SEM = " + str(semOfMeansC2_12[7]))
print("Val 3, 6, 12 " + " Hybrid mean = " + str(meanOfMeansHybrid_12[7]) + " Hybrid SEM = " + str(semOfMeansHybrid_12[7]))
```

In [ ]:
```
print("Val 1, 6, 9 " + " C2 mean = " + str(meanOfMeansC2_12[8]) + " C2 SEM = " + str(semOfMeansC2_12[8]))
print("Val 1, 6, 9 " + " Hybrid mean = " + str(meanOfMeansHybrid_12[8]) + " Hybrid SEM = " + str(semOfMeansHybrid_12[8]))
```

In [ ]:
```
print("Val 2, 7, 10 " + " C2 mean = " + str(meanOfMeansC2_12[9]) + " C2 SEM = " + str(semOfMeansC2_12[9]))
print("Val 2, 7, 10 " + " Hybrid mean = " + str(meanOfMeansHybrid_12[9]) + " Hybrid SEM = " + str(semOfMeansHybrid_12[9]))
```

In [ ]:
```
import plotly.graph_objects as go
import kaleido

validationSets = ['Val 1, 2, 3','Val 4, 5, 6', 'Val 7, 8, 9', 'Val 10, 11, 12',
                  'Val 1 4 7', 'Val 5 8 10', 'Val 2 9 11', 'Val 3 6 12',
                  'Val 1 6 9', 'Val 2 7 10', 'Val 3 8 11', 'Val 4 5 12']

fig = go.Figure()
fig.add_trace(go.Scatter(
    x=validationSets,
    y=meanOfMeansC2_12,
    error_y=dict(
                type='data',
                symmetric=True,
                color='black',
                thickness=1,
                width=8,
                array=semOfMeansC2_12),
    name='C2',
    mode='markers',
    marker=dict(color="#00429d", size=8)
))
fig.add_trace(go.Scatter(
    x=validationSets,
    y=meanOfMeansHybrid_12,
    error_y=dict(
                type='data',
                symmetric=True,
                color='black',
                thickness=1,
                width=8,
                array=semOfMeansHybrid_12),
    name='Hybrid',
    mode='markers',
    marker=dict(color="#93003a", size=8)

))

title="Means & SEM - 13 Features - Binary Crystals Crystal Rating (C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024)"

fig.update_layout(xaxis_tickangle=-45,
                    plot_bgcolor="#fff",
                    title=title,
                    xaxis_title="Validation Set",
                    yaxis_title="Mean",
                font=dict(
                    family="Helvetica",
                    size=9,
                    color="Black"))
fig.update_yaxes(showgrid=True, gridwidth=0.5, gridcolor='LightPink', minor_griddash="dot")
fig.update_layout(title_pad_l=400,
                    title_pad_r=400,
                    title={
                    'text': title,
                    'y':0.9,
                    'x':0.5,
                    'xanchor': 'center',
                    'yanchor': 'top'})

fig.show()

fig.write_image(path + "/" + "Means and SEM - C2 vs Hybrid Networks.png")
```