In [1]:

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

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

### 13 Features
#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 - 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"

### 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"

### KES 2022 - 12 runs of 12 alternating validation sets
#path =  root + "/13 Features - Binary Crystals/\
#Open_Folders_WC_25_04_22_Results from 13 features_ 12 Sets of 12 Runs"

### 13 Re-rated expert features (ex-additional images)
#path =  root + "/Re-rated expertFeatures - 13 - Binary/\
#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"
```

In [2]:

```
keywordConfusion = '*Confusion*'
all_Confusion = []

#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)
```

In [3]:

```
list_of_12_of_12 = [all_Confusion[i:i+12] for i in range(0, len(all_Confusion),12)]
```

In [ ]:

```
run_1 = list_of_12_of_12[0]
run_2 = list_of_12_of_12[1]
run_3 = list_of_12_of_12[2]
run_4 = list_of_12_of_12[3]
run_5 = list_of_12_of_12[4]
run_6 = list_of_12_of_12[5]
run_7 = list_of_12_of_12[6]
run_8 = list_of_12_of_12[7]
run_9 = list_of_12_of_12[8]
run_10 = list_of_12_of_12[9]
run_11 = list_of_12_of_12[10]
run_12 = list_of_12_of_12[11]
```

In [ ]:

```
run_1_Df = []
run_2_Df = []
run_3_Df = []
run_4_Df = []
run_5_Df = []
run_6_Df = []
run_7_Df = []
run_8_Df = []
run_9_Df = []
run_10_Df = []
run_11_Df = []
run_12_Df = []
```

In [ ]:

```
run_1_Acc_Hybrid = pd.DataFrame()
run_2_Acc_Hybrid = pd.DataFrame()
run_3_Acc_Hybrid = pd.DataFrame()
run_4_Acc_Hybrid = pd.DataFrame()
run_5_Acc_Hybrid = pd.DataFrame()
run_6_Acc_Hybrid = pd.DataFrame()
run_7_Acc_Hybrid = pd.DataFrame()
run_8_Acc_Hybrid = pd.DataFrame()
run_9_Acc_Hybrid = pd.DataFrame()
run_10_Acc_Hybrid = pd.DataFrame()
run_11_Acc_Hybrid = pd.DataFrame()
run_12_Acc_Hybrid = pd.DataFrame()
```

In [ ]:

```
run_1_Acc_C2 = pd.DataFrame()
run_2_Acc_C2  = pd.DataFrame()
run_3_Acc_C2  = pd.DataFrame()
run_4_Acc_C2  = pd.DataFrame()
run_5_Acc_C2  = pd.DataFrame()
run_6_Acc_C2  = pd.DataFrame()
run_7_Acc_C2  = pd.DataFrame()
run_8_Acc_C2  = pd.DataFrame()
run_9_Acc_C2  = pd.DataFrame()
run_10_Acc_C2  = pd.DataFrame()
run_11_Acc_C2  = pd.DataFrame()
run_12_Acc_C2  = pd.DataFrame()
```

In [5]:

```
run_1_Acc_Unconstrained = pd.DataFrame()
run_2_Acc_Unconstrained  = pd.DataFrame()
run_3_Acc_Unconstrained  = pd.DataFrame()
run_4_Acc_Unconstrained  = pd.DataFrame()
run_5_Acc_Unconstrained  = pd.DataFrame()
run_6_Acc_Unconstrained  = pd.DataFrame()
run_7_Acc_Unconstrained  = pd.DataFrame()
run_8_Acc_Unconstrained  = pd.DataFrame()
run_9_Acc_Unconstrained  = pd.DataFrame()
run_10_Acc_Unconstrained  = pd.DataFrame()
run_11_Acc_Unconstrained  = pd.DataFrame()
run_12_Acc_Unconstrained  = pd.DataFrame()
```

In [6]:

```
def listToDf (run, runDF):
    for list in run:
        df = pd.read_csv(list, header=None)
        runDF.append(df)
    return runDF
```

In [7]:

```
run_1_Acc_Df = listToDf(run_1,run_1_Df)
run_2_Acc_Df = listToDf(run_2,run_2_Df)
run_3_Acc_Df = listToDf(run_3,run_3_Df)
run_4_Acc_Df = listToDf(run_4,run_4_Df)
run_5_Acc_Df = listToDf(run_5,run_5_Df)
run_6_Acc_Df = listToDf(run_6,run_6_Df)
run_7_Acc_Df = listToDf(run_7,run_7_Df)
run_8_Acc_Df = listToDf(run_8,run_8_Df)
run_9_Acc_Df = listToDf(run_9,run_9_Df)
run_10_Acc_Df = listToDf(run_10,run_10_Df)
run_11_Acc_Df = listToDf(run_11,run_11_Df)
run_12_Acc_Df = listToDf(run_12,run_12_Df)
```

In [8]:

```
def splitAccuraciesToDfC2(constrainedC2RunAcc, run):
    for df in run:
        a = df.iloc[12,1:2]
        constrainedC2RunAcc = pd.concat([constrainedC2RunAcc, a], axis=0,ignore_index=True)
    return constrainedC2RunAcc
```

In [9]:

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

In [10]:

```
# Unconstrained post-KES 2022
# Coordinates in csv are for C2 for KES 2022 data set
def splitAccuraciesToDfUnconstrained(unconstrainedNetworkRunAcc, run):
    for df in run:
        a = df.iloc[12,21:22]
        unconstrainedNetworkRunAcc = pd.concat([unconstrainedNetworkRunAcc, a], axis=0,ignore_index=True)
    return unconstrainedNetworkRunAcc
```

In [11]:

```
run_1_Acc_Hybrid = splitAccuraciesToDfHybrid(run_1_Acc_Hybrid, run_1_Acc_Df)
run_2_Acc_Hybrid = splitAccuraciesToDfHybrid(run_2_Acc_Hybrid, run_2_Acc_Df)
run_3_Acc_Hybrid = splitAccuraciesToDfHybrid(run_3_Acc_Hybrid, run_3_Acc_Df)
run_4_Acc_Hybrid = splitAccuraciesToDfHybrid(run_4_Acc_Hybrid, run_4_Acc_Df)
run_5_Acc_Hybrid = splitAccuraciesToDfHybrid(run_5_Acc_Hybrid, run_5_Acc_Df)
run_6_Acc_Hybrid = splitAccuraciesToDfHybrid(run_6_Acc_Hybrid, run_6_Acc_Df)
run_7_Acc_Hybrid = splitAccuraciesToDfHybrid(run_7_Acc_Hybrid, run_7_Acc_Df)
run_8_Acc_Hybrid = splitAccuraciesToDfHybrid(run_8_Acc_Hybrid, run_8_Acc_Df)
run_9_Acc_Hybrid = splitAccuraciesToDfHybrid(run_9_Acc_Hybrid, run_9_Acc_Df)
run_10_Acc_Hybrid = splitAccuraciesToDfHybrid(run_10_Acc_Hybrid, run_10_Acc_Df)
run_11_Acc_Hybrid = splitAccuraciesToDfHybrid(run_11_Acc_Hybrid, run_11_Acc_Df)
run_12_Acc_Hybrid = splitAccuraciesToDfHybrid(run_12_Acc_Hybrid, run_12_Acc_Df)
```

In [12]:

```
run_1_Acc_C2 = splitAccuraciesToDfC2(run_1_Acc_C2, run_1_Acc_Df)
run_2_Acc_C2 = splitAccuraciesToDfC2(run_2_Acc_C2, run_2_Acc_Df)
run_3_Acc_C2 = splitAccuraciesToDfC2(run_3_Acc_C2, run_3_Acc_Df)
run_4_Acc_C2 = splitAccuraciesToDfC2(run_4_Acc_C2, run_4_Acc_Df)
run_5_Acc_C2 = splitAccuraciesToDfC2(run_5_Acc_C2, run_5_Acc_Df)
run_6_Acc_C2 = splitAccuraciesToDfC2(run_6_Acc_C2, run_6_Acc_Df)
run_7_Acc_C2 = splitAccuraciesToDfC2(run_7_Acc_C2, run_7_Acc_Df)
run_8_Acc_C2 = splitAccuraciesToDfC2(run_8_Acc_C2, run_8_Acc_Df)
run_9_Acc_C2 = splitAccuraciesToDfC2(run_9_Acc_C2, run_9_Acc_Df)
run_10_Acc_C2 = splitAccuraciesToDfC2(run_10_Acc_C2, run_10_Acc_Df)
run_11_Acc_C2 = splitAccuraciesToDfC2(run_11_Acc_C2, run_11_Acc_Df)
run_12_Acc_C2 = splitAccuraciesToDfC2(run_12_Acc_C2, run_12_Acc_Df)
```

In [13]:

```
run_1_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_1_Acc_Unconstrained, run_1_Acc_Df)
run_2_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_2_Acc_Unconstrained, run_2_Acc_Df)
run_3_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_3_Acc_Unconstrained, run_3_Acc_Df)
run_4_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_4_Acc_Unconstrained, run_4_Acc_Df)
run_5_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_5_Acc_Unconstrained, run_5_Acc_Df)
run_6_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_6_Acc_Unconstrained, run_6_Acc_Df)
run_7_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_7_Acc_Unconstrained, run_7_Acc_Df)
run_8_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_8_Acc_Unconstrained, run_8_Acc_Df)
run_9_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_9_Acc_Unconstrained, run_9_Acc_Df)
run_10_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_10_Acc_Unconstrained, run_10_Acc_Df)
run_11_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_11_Acc_Unconstrained, run_11_Acc_Df)
run_12_Acc_Unconstrained = splitAccuraciesToDfUnconstrained(run_12_Acc_Unconstrained, run_12_Acc_Df)
```

In [14]:

```
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 [15]:

```
run_1_Acc_Hybrid_MeanStdSem = meanValStdSem(run_1_Acc_Hybrid)
run_2_Acc_Hybrid_MeanStdSem = meanValStdSem(run_2_Acc_Hybrid)
run_3_Acc_Hybrid_MeanStdSem = meanValStdSem(run_3_Acc_Hybrid)
run_4_Acc_Hybrid_MeanStdSem = meanValStdSem(run_4_Acc_Hybrid)
run_5_Acc_Hybrid_MeanStdSem = meanValStdSem(run_5_Acc_Hybrid)
run_6_Acc_Hybrid_MeanStdSem = meanValStdSem(run_6_Acc_Hybrid)
run_7_Acc_Hybrid_MeanStdSem = meanValStdSem(run_7_Acc_Hybrid)
run_8_Acc_Hybrid_MeanStdSem = meanValStdSem(run_8_Acc_Hybrid)
run_9_Acc_Hybrid_MeanStdSem = meanValStdSem(run_9_Acc_Hybrid)
run_10_Acc_Hybrid_MeanStdSem = meanValStdSem(run_10_Acc_Hybrid)
run_11_Acc_Hybrid_MeanStdSem = meanValStdSem(run_11_Acc_Hybrid)
run_12_Acc_Hybrid_MeanStdSem = meanValStdSem(run_12_Acc_Hybrid)
```

In [16]:

```
all_hybrid_means = [run_1_Acc_Hybrid_MeanStdSem [0], run_2_Acc_Hybrid_MeanStdSem [0], run_3_Acc_Hybrid_MeanStdSem [0],
                    run_4_Acc_Hybrid_MeanStdSem [0], run_5_Acc_Hybrid_MeanStdSem [0], run_6_Acc_Hybrid_MeanStdSem [0],
                    run_7_Acc_Hybrid_MeanStdSem [0], run_8_Acc_Hybrid_MeanStdSem [0], run_9_Acc_Hybrid_MeanStdSem [0],
                    run_10_Acc_Hybrid_MeanStdSem [0], run_11_Acc_Hybrid_MeanStdSem [0], run_12_Acc_Hybrid_MeanStdSem [0]
                   ]
```

In [17]:

```
all_hybrid_means = pd.DataFrame(all_hybrid_means)
all_Hybrid_MeanStdSem = meanValStdSem(all_hybrid_means)
print(all_Hybrid_MeanStdSem)
```

```
(87.59259583333333, 2.9448213347963117, 0.8500966951800019)
```

In [18]:

```
run_1_Acc_C2_MeanStdSem = meanValStdSem(run_1_Acc_C2)
run_2_Acc_C2_MeanStdSem = meanValStdSem(run_2_Acc_C2)
run_3_Acc_C2_MeanStdSem = meanValStdSem(run_3_Acc_C2)
run_4_Acc_C2_MeanStdSem = meanValStdSem(run_4_Acc_C2)
run_5_Acc_C2_MeanStdSem = meanValStdSem(run_5_Acc_C2)
run_6_Acc_C2_MeanStdSem = meanValStdSem(run_6_Acc_C2)
run_7_Acc_C2_MeanStdSem = meanValStdSem(run_7_Acc_C2)
run_8_Acc_C2_MeanStdSem = meanValStdSem(run_8_Acc_C2)
run_9_Acc_C2_MeanStdSem = meanValStdSem(run_9_Acc_C2)
run_10_Acc_C2_MeanStdSem = meanValStdSem(run_10_Acc_C2)
run_11_Acc_C2_MeanStdSem = meanValStdSem(run_11_Acc_C2)
run_12_Acc_C2_MeanStdSem = meanValStdSem(run_12_Acc_C2)
```

In [19]:

```
all_C2_means =[run_1_Acc_C2_MeanStdSem [0], run_2_Acc_C2_MeanStdSem [0], run_3_Acc_C2_MeanStdSem [0],
               run_4_Acc_C2_MeanStdSem [0], run_5_Acc_C2_MeanStdSem [0], run_6_Acc_C2_MeanStdSem [0],
               run_7_Acc_C2_MeanStdSem [0], run_8_Acc_C2_MeanStdSem [0], run_9_Acc_C2_MeanStdSem [0],
               run_10_Acc_C2_MeanStdSem [0], run_11_Acc_C2_MeanStdSem [0], run_12_Acc_C2_MeanStdSem [0]
              ]
```

In [20]:

```
all_C2_means = pd.DataFrame(all_C2_means)
all_C2_MeanStdSem = meanValStdSem(all_C2_means)
print(all_C2_MeanStdSem)
```

```
(84.62963125, 4.030702572628562, 1.1635636076652087)
```

In [21]:

```
run_1_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_1_Acc_Unconstrained)
run_2_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_2_Acc_Unconstrained)
run_3_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_3_Acc_Unconstrained)
run_4_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_4_Acc_Unconstrained)
run_5_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_5_Acc_Unconstrained)
run_6_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_6_Acc_Unconstrained)
run_7_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_7_Acc_Unconstrained)
run_8_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_8_Acc_Unconstrained)
run_9_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_9_Acc_Unconstrained)
run_10_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_10_Acc_Unconstrained)
run_11_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_11_Acc_Unconstrained)
run_12_Acc_Unconstrained_MeanStdSem = meanValStdSem(run_12_Acc_Unconstrained)
```

In [22]:

```
all_Unconstrained_means =[run_1_Acc_Unconstrained_MeanStdSem [0], run_2_Acc_Unconstrained_MeanStdSem [0],
                          run_3_Acc_Unconstrained_MeanStdSem [0], run_4_Acc_Unconstrained_MeanStdSem [0],
                          run_5_Acc_Unconstrained_MeanStdSem [0], run_6_Acc_Unconstrained_MeanStdSem [0],
                          run_7_Acc_Unconstrained_MeanStdSem [0], run_8_Acc_Unconstrained_MeanStdSem [0],
                          run_9_Acc_Unconstrained_MeanStdSem [0], run_10_Acc_Unconstrained_MeanStdSem [0],
                          run_11_Acc_Unconstrained_MeanStdSem [0], run_12_Acc_Unconstrained_MeanStdSem [0]
                         ]
```

In [23]:

```
# This is the C2 model results for KES 2022
# This is unconstrained post-KES 2022

all_Unconstrained_means = pd.DataFrame(all_Unconstrained_means)
all_Unconstrained_MeanStdSem = meanValStdSem(all_Unconstrained_means)
print(all_Unconstrained_MeanStdSem)
```

```
(87.15278124999999, 4.225095966331663, 1.219680146756794)
```