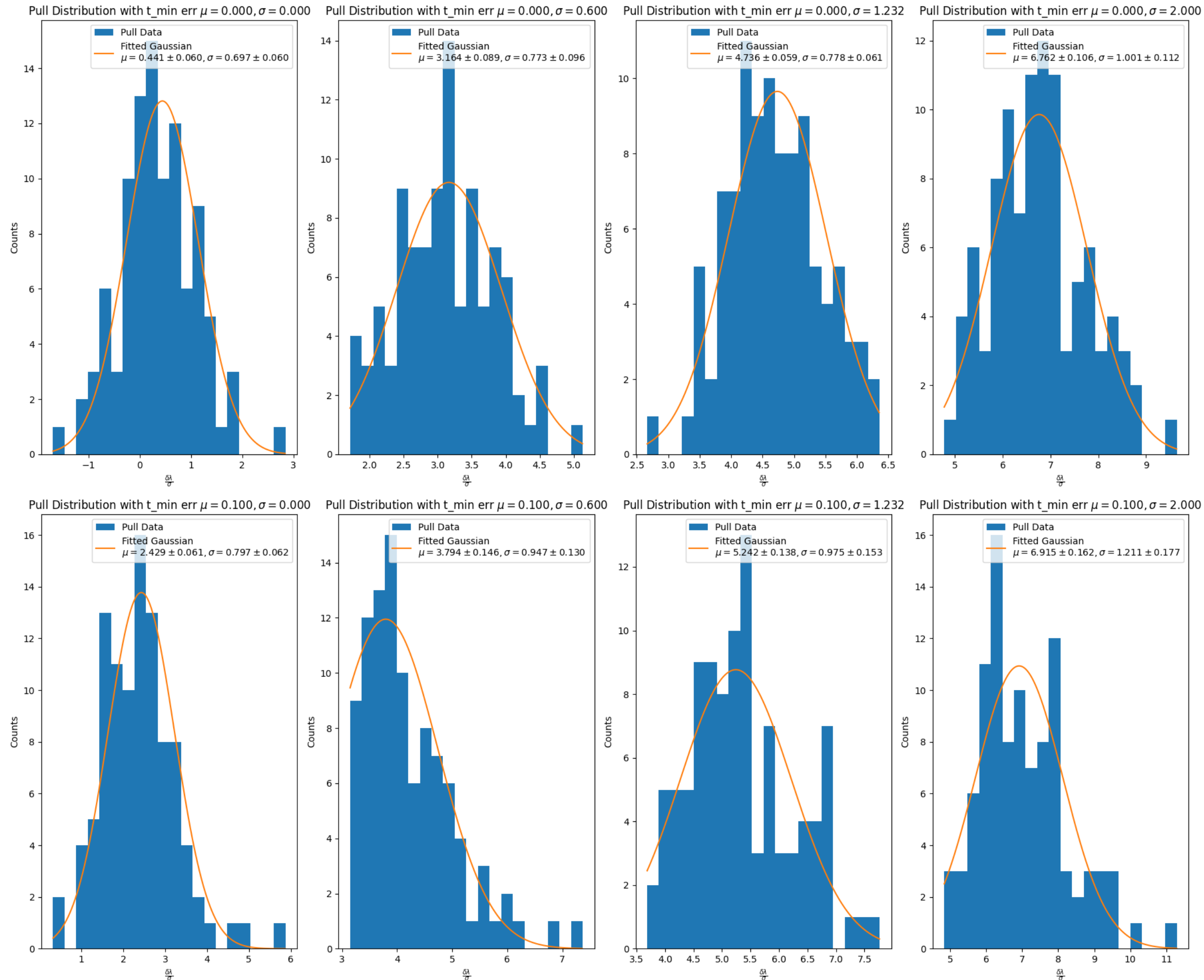




Quantum Decoherence - Status Update



UNIVERSITY
of HAWAII[®]
MĀNOA



```
dim = 2
theolam = 0.5
N = 10_000
mintBins = 3
dtBins = 10
showplots = False
debug = 0

tau = 1.5
dm = 0.471

iters = 100
PullDist = np.zeros(iters)

#muList = [-0.185769, -0.1, 0, 0.1, 0.185769]
muList = [0.0, 0.1]
sigmaList = [0, 0.6, 1.23244, 2]
#sigmaList = [0.5, 1.0, 1.5, 2.0]
fig, ax = plt.subplots(len(muList), len(sigmaList), figsize=[15, 18]) #

for row, mu in enumerate(muList):
    for col, sigma in enumerate(sigmaList):
        PullDist = np.zeros(iters)

        for i in range(iters):
            t1 = np.random.exponential(scale=tau, size=N) * 1.04
            t2 = np.random.exponential(scale=tau, size=N) * 1.04

            deltaT = t2 - t1
            SFProb = CumulativeProb(keyword='NSF', t1=t1, t2=t2, lam=theolam)
            LogicalComparison = np.random.uniform(0, 1, N)

            SameFlavorCM = SFProb > LogicalComparison

            OppositeFlavorCM = ~SameFlavorCM
            MinimumTimeCM = np.minimum(t1, t2)

            MinimumTimeCM_Error = 0.0

            MinimumTimeCM_withGauss, MinimumTimeCM_withGaussError = AddGaussianNoise(MinimumTimeCM, MinimumTimeCM_Error, mu=mu, sigma=sigma)

            dt, sf, of, mint, epb, ProbError, dtError, minTErr = BinData(NumOfdtBins = dtBins, NumOfMinTBins = mintBins, deltaT = deltaT, SameFlavor = SameFlavorCM,
                                OppositeFlavor = OppositeFlavorCM, MinimumTime = MinimumTimeCM_withGauss, MinimumTimeError = MinimumTimeCM_withGaussError, dm = dm,
                                showplots = showplots, debug = debug)

            # print(f'before errors are added mint is {mint} \n and minTErrors are {minTErr}')
            #mint, minTErr = AddGaussianNoise(mint, minTErr, mu=mu, sigma=sigma)
            # print(f'after errors are added mint is {mint} \n and minTErrors are {minTErr}')
            Errors = ProbError[:2]
            SFLam, SFLamErr, P = FitDataWithROOT(deltaT = dt, Prob = sf, MinTime = mint, yError = Errors, TheoreticalLambda=theolam,
                                                  dtError=dtError, minTErr=minTErr, flavor = 'Same', dim = dim, dm=dm, returnplot=True)

            if i == 0:
                ShowRootGraphs(P, numofPlots_mt=mintBins, numofPlots_dt=dtBins, dim=dim)

            PullDist[i] = (theolam - SFLam) / np.abs(SFLamErr)
```