

Week 19

0. Table of Contents

1. Introduction
2. Analysis Update
3. Higher Number Interpolation
4. Preliminary Confusion Code
5. Improved Full Tumbling Rate Analysis (more rolling, more epochs)
6. Running Same Parameters Multiple Times
7. Auto-Correlation Function

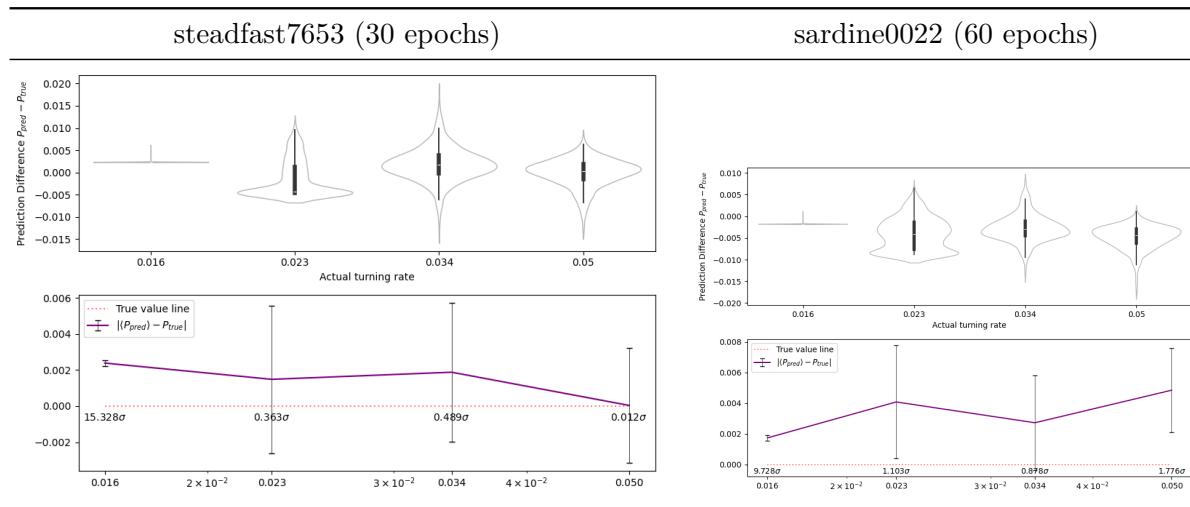
1. Introduction

This week consists of exploring higher range parameters in order to consolidate the results in previous weeks.

2. Analysis Update

We have finally fixed GPU tensorflow, which greatly accelerates model training.

steadfast7653: $P_t \in \{0.016, 0.023, 0.034, 0.050\}$, $\rho = 0.25$, **30 epochs** ; **sardine0022:** **same, same, 60 epochs**



Prediction means and standard deviations for steadfast7653.

Actual value 0.016: Average = 0.01838 +- 0.00016; Expected value within 15.328 stdevs of mean

Actual value 0.023: Average = 0.02152 +- 0.00409; Expected value within 0.363 stdevs of mean

Actual value 0.034: Average = 0.03588 +- 0.00384; Expected value within 0.489 stdevs of mean

Actual value 0.05: Average = 0.04996 +- 0.00318; Expected value within 0.012 stdevs of mean

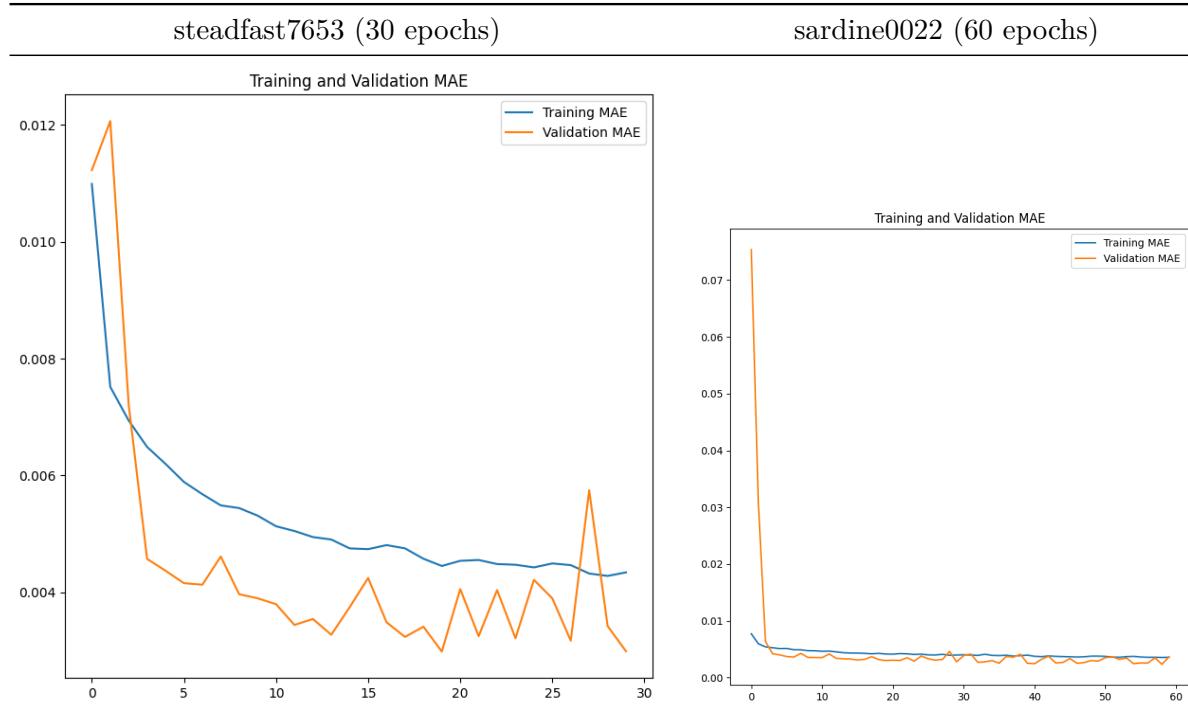
Prediction means and standard deviations for sardine0022.

Actual value 0.016: Average = 0.01427 +- 0.00018; Expected value within 9.728 stdevs of mean

Actual value 0.023: Average = 0.01892 +- 0.00370; Expected value within 1.103 stdevs of mean

Actual value 0.034: Average = 0.03128 +- 0.00310; Expected value within 0.878 stdevs of mean

Actual value 0.05: Average = 0.04516 +- 0.00273; Expected value within 1.776 stdevs of mean

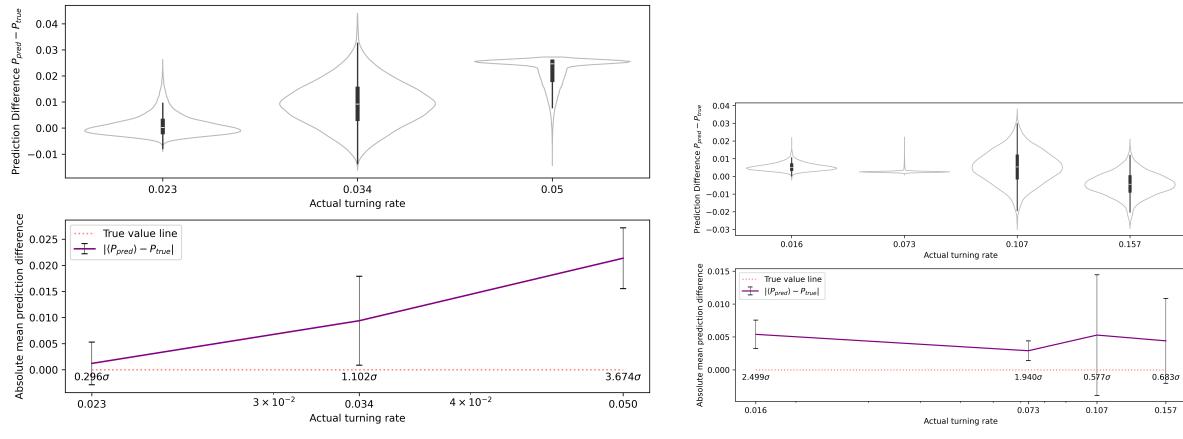


3. Higher Number Interpolation

outstretched4188 (antenna4149): $P_{val} \in \{0.023, 0.034, 0.050\}$, $P_{train} \in \{0.016, 0.073, 0.107, 0.157\}$, $\rho = 0.25$, **30 epochs**

outstretched4188 (evaluation)

antenna4149 (training)



Prediction means and standard deviations for outstretched4188.

Actual value 0.023: Average = 0.02421 +- 0.00409; Expected value within 0.296 stdevs of mean

Actual value 0.034: Average = 0.04339 +- 0.00852; Expected value within 1.102 stdevs of mean

Actual value 0.05: Average = 0.07137 +- 0.00582; Expected value within 3.674 stdevs of mean

Prediction means and standard deviations for antenna4149.

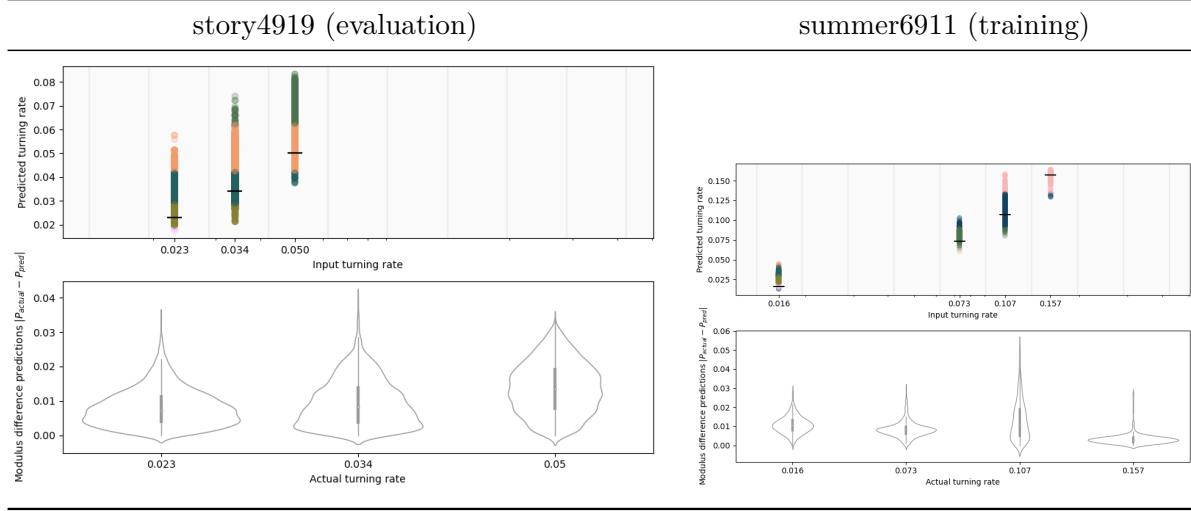
Actual value 0.016: Average = 0.02139 +- 0.00216; Expected value within 2.499 stdevs of mean

Actual value 0.073: Average = 0.07590 +- 0.00150; Expected value within 1.940 stdevs of mean

Actual value 0.107: Average = 0.11229 +- 0.00916; Expected value within 0.577 stdevs of mean

Actual value 0.157: Average = 0.15259 +- 0.00645; Expected value within 0.683 stdevs of mean

Compare to equivalent case from [last week](#):



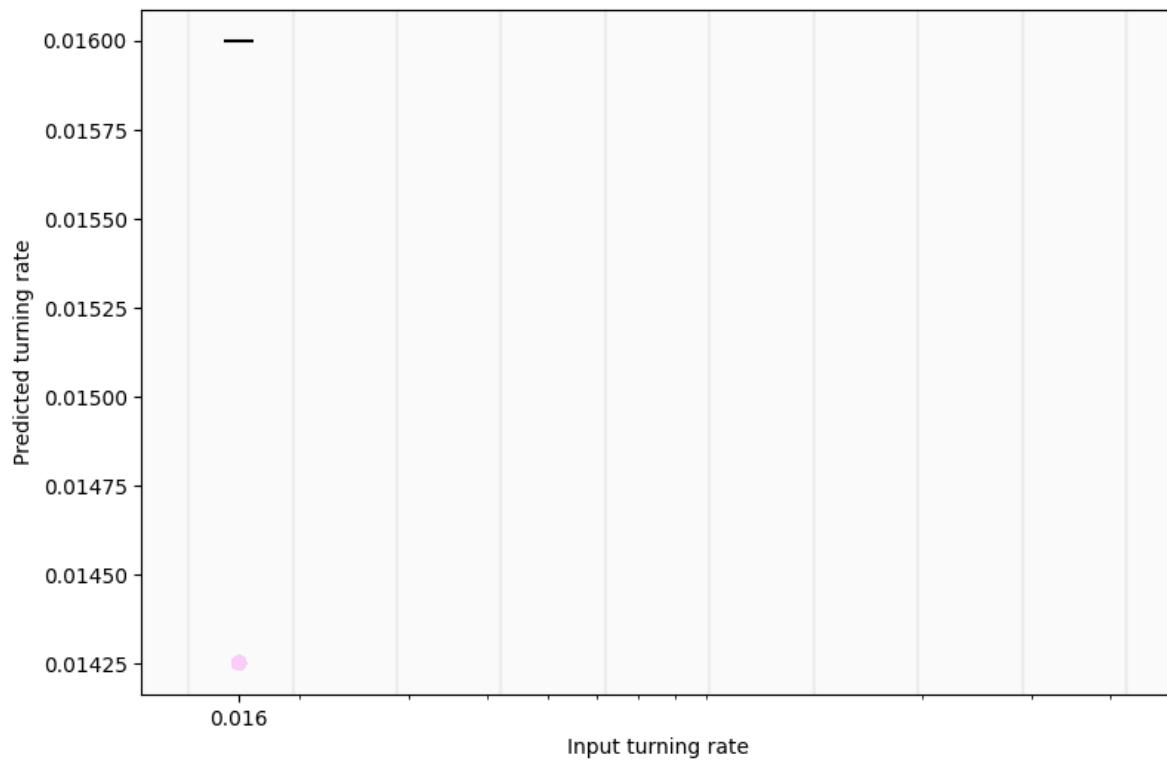
4. Preliminary Confusion Code

```

img = fin[key] [:]
for i in range(128):
    for j in range(128):
        if img[i,j] > 0:
            chance=float(random.randint(1,100))/100
            if chance <= 0.05:
                while (True):
                    x = random.randint(1,4)
                    if x != img[i,j]:
                        break
                img[i,j] = x
img = img.astype(float)/4.0

```

less5622: $P_{tumbke} = 0.016$, $\rho = 0.25$, **60 epochs, 4000 (0.2) snapshots**

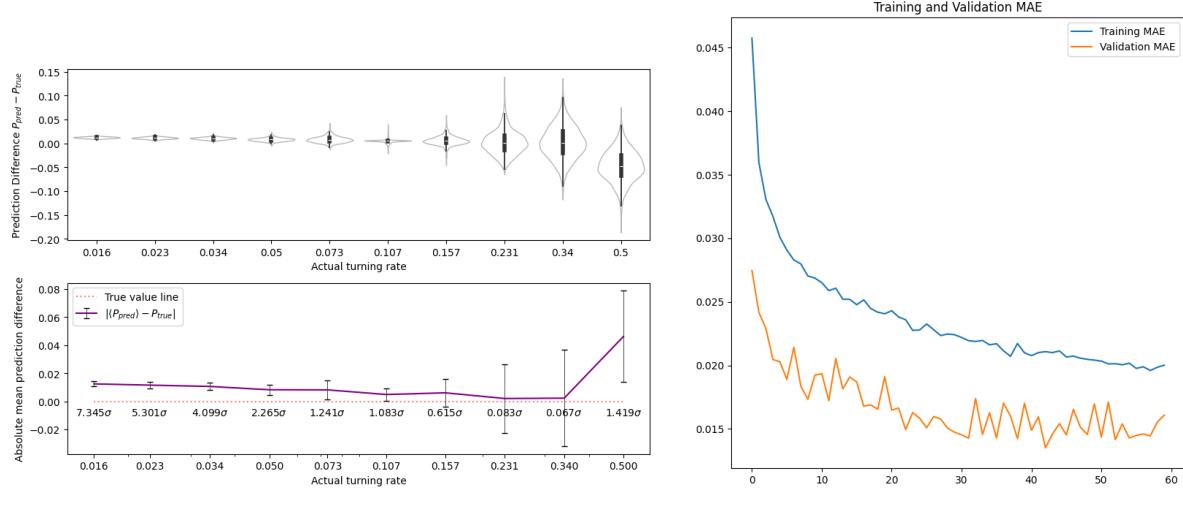


5. Improved Full Tumbling Rate Analysis (more rolling, more epochs)

stag0149: $P_{tumble} \in \{0.016, 0.023, 0.034, 0.050, 0.073, 0.107, 0.157, 0.231, 0.340, 0.500\}$, $\rho = 0.25$, **60 epochs, 80000 (0.2) snapshots**

stag0149 Predictions

stag0149 Loss

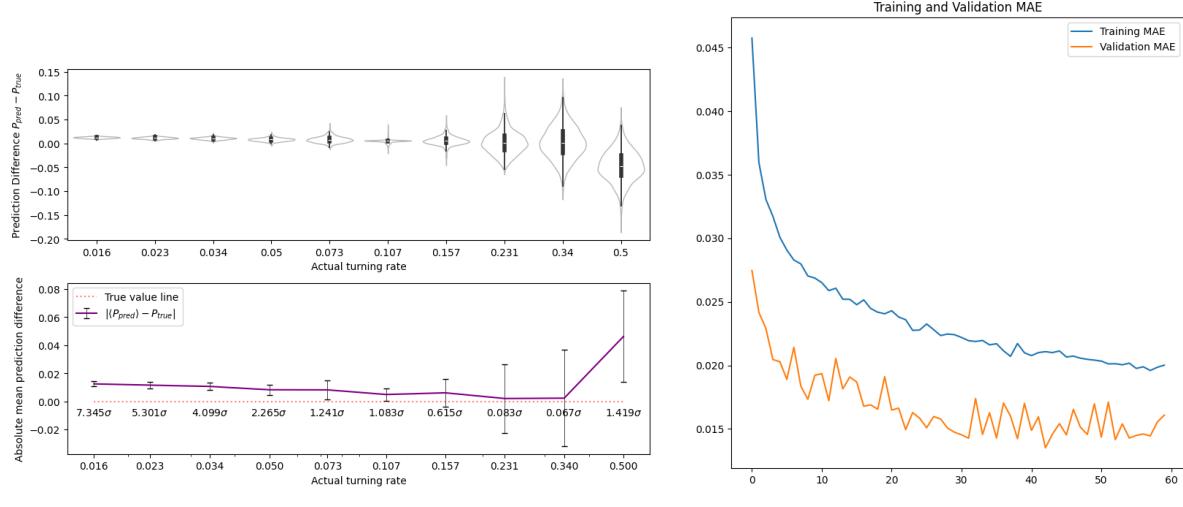


6. Running Same Parameters Multiple Times

stag0149, leaflet5121 and branch3151 are different runs of the same parameters (see Section 5).

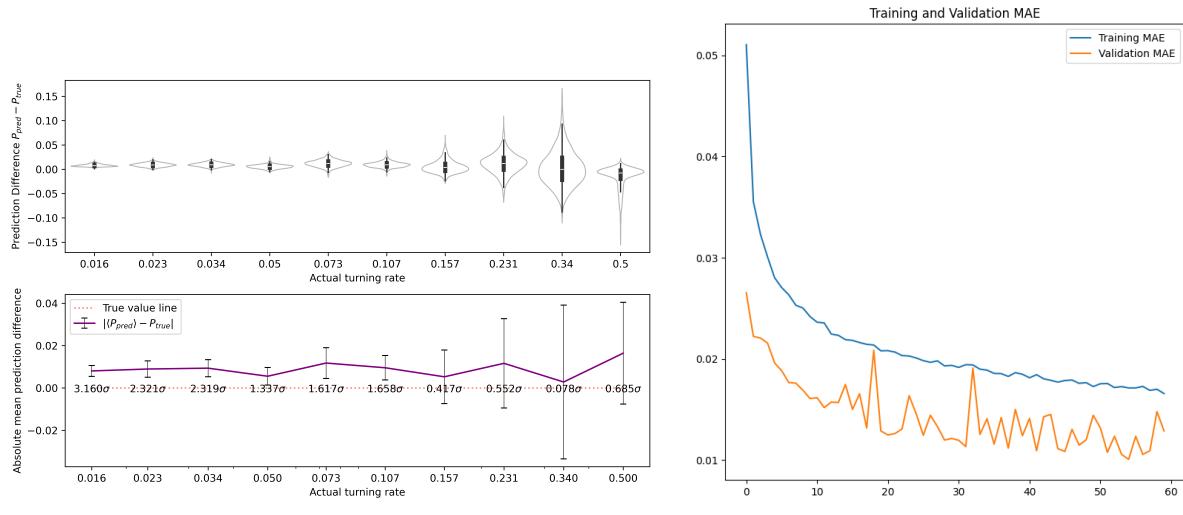
stag0149 Predictions

stag0149 Loss



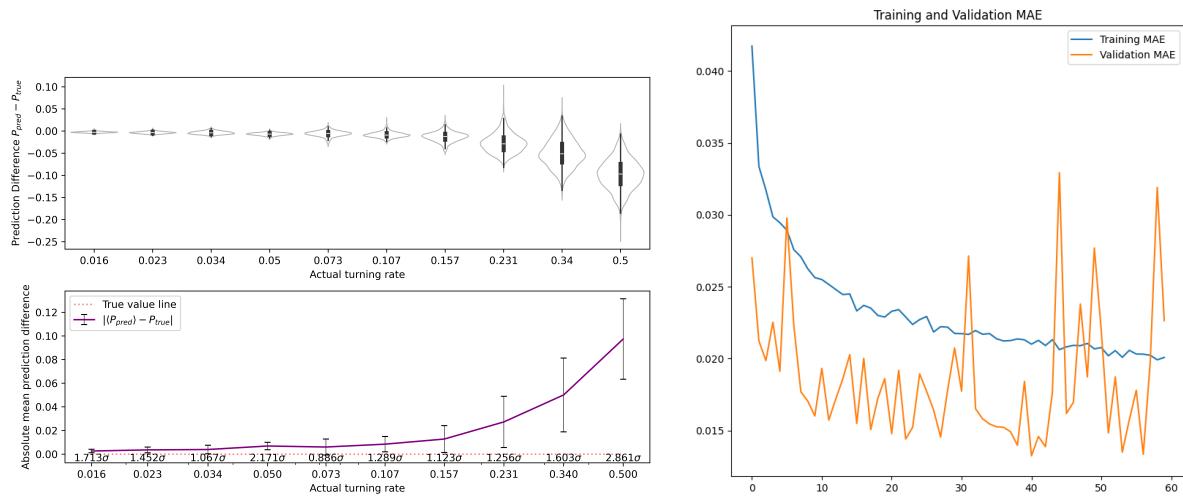
leaflet5121 Predictions

leaflet5121 Loss



branch3151 Predictions

branch3151 Loss



7. Auto-Correlation Function

```

def overlap(traj,i,j):
    N = traj[i][traj[i]>0].shape[0]
    return ((traj[i]>0)*(traj[j]>0)).sum()/N

def acf_analysis (tumble,density,lags,data):

```

```

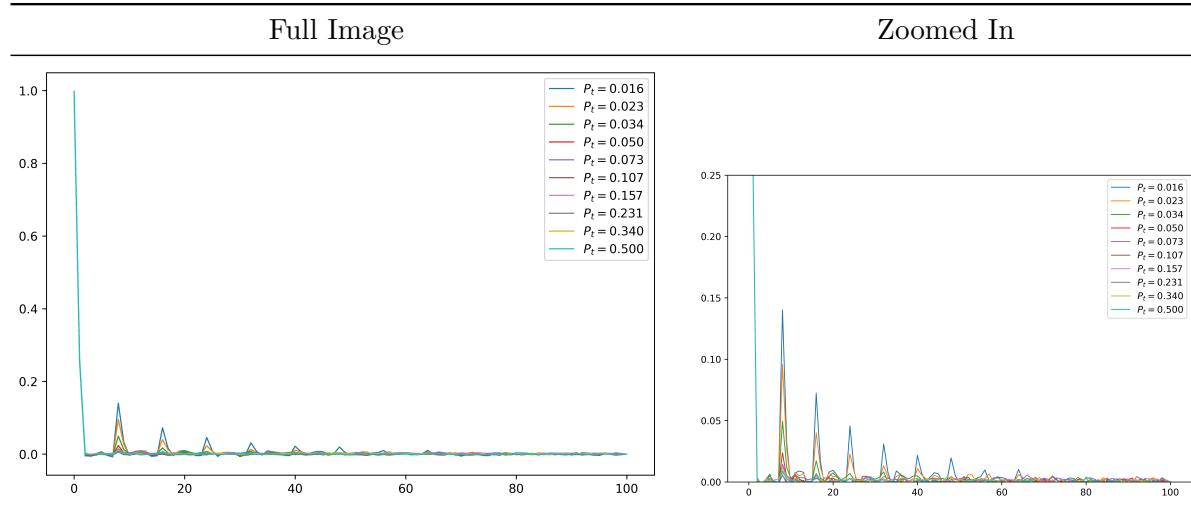
acf = []
for i in range(0,500,10):
    a = [overlap(data,i,i+lag) for lag in lags]
    acf.append(a)

acf = np.asarray(acf).mean(axis=0)
acf = acf-acf[-1]

acf/= acf.ptp()
return acf

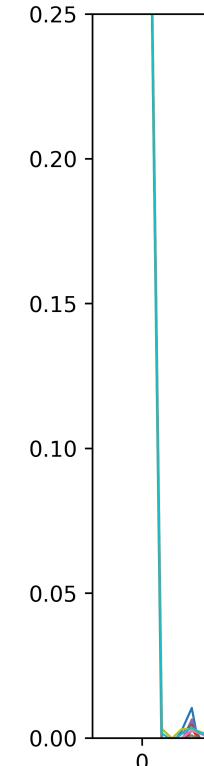
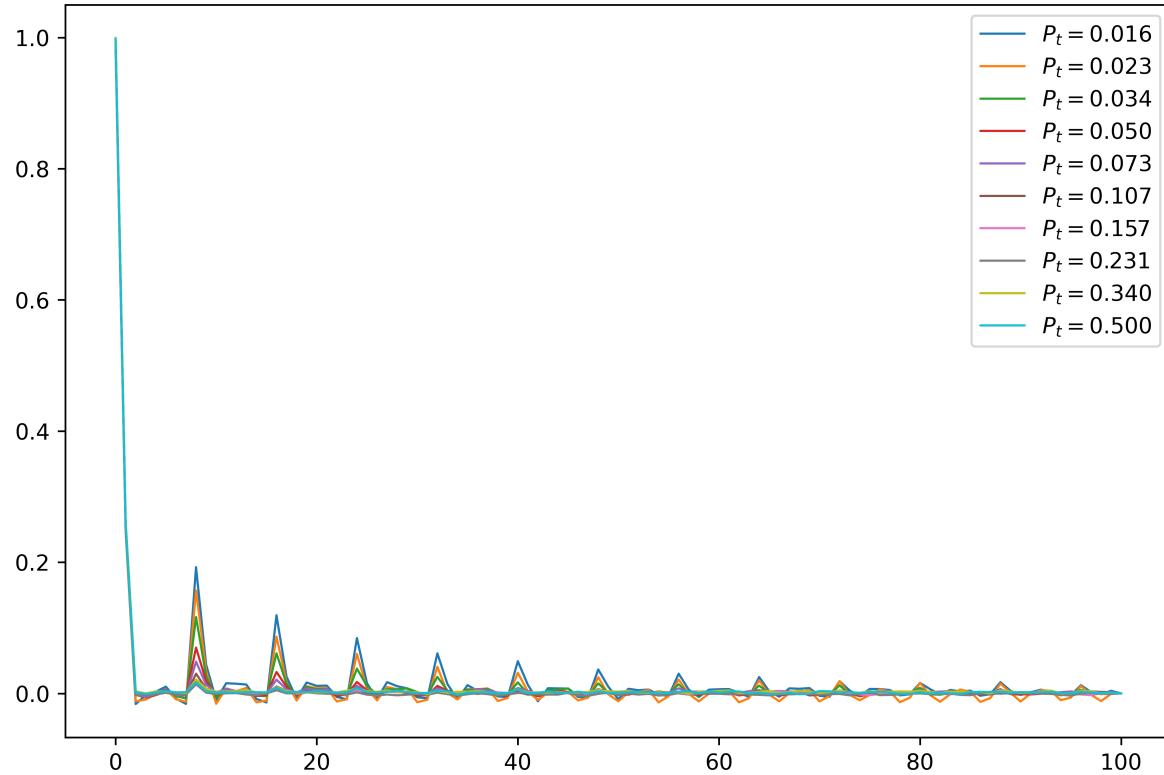
```

$$\rho = 0.15$$



$$\rho = 0.25$$

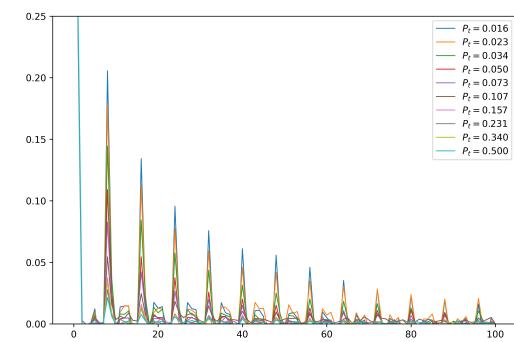
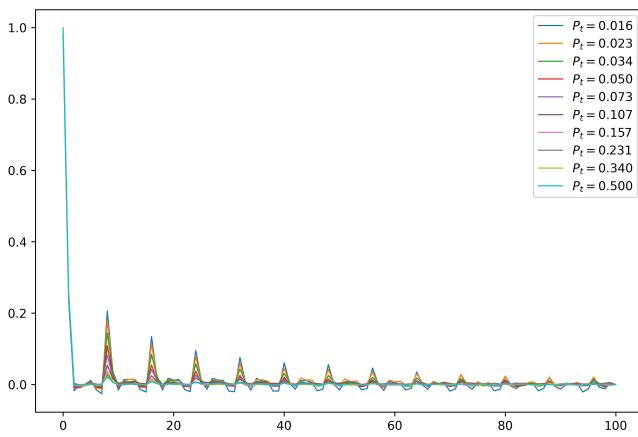
Full Image



$\rho = 0.35$

Full Image

Zoomed In



$$\rho = 0.5$$

