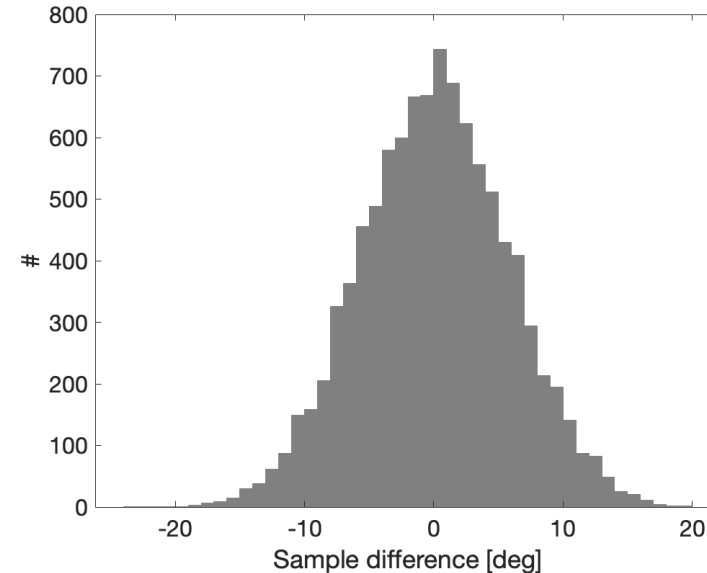
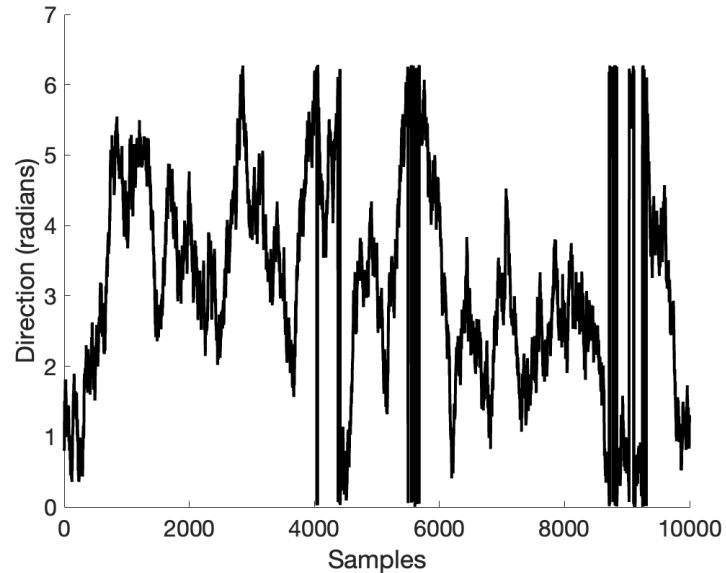


Random walk implementation



```
% Step size in polar space
```

```
polar_step_size = 0.1;
```

```
% Initial stimulus direction
```

```
stimulus_direction = rand * 2 * pi;
```

```
% Update stimulus direction with random step
```

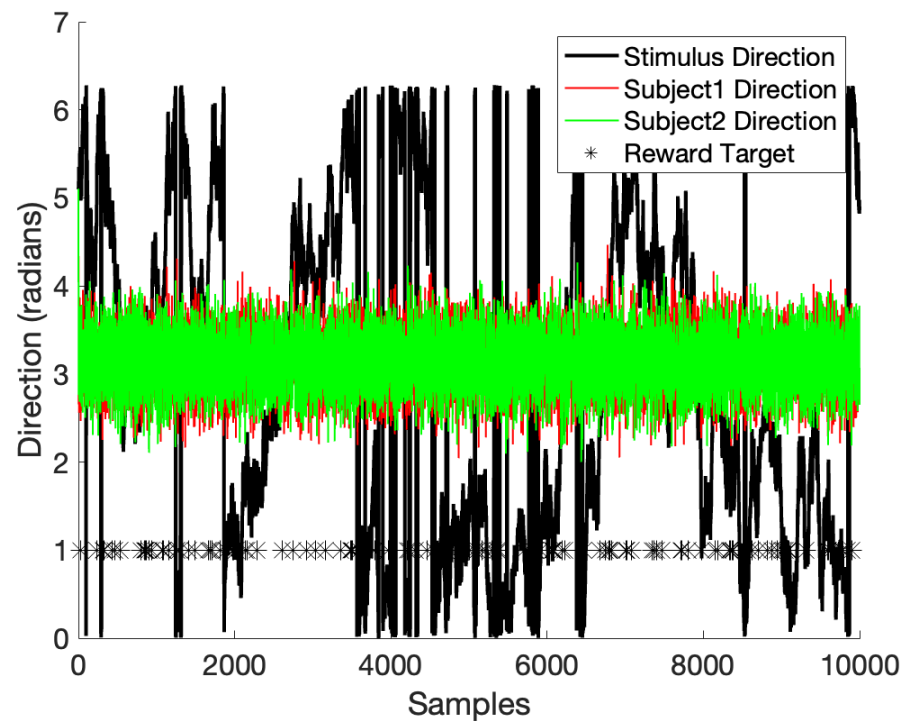
```
stimulus_direction(iStep) = stimulus_direction(iStep-1) + randn * polar_step_size;
```

Subject tracking simulation

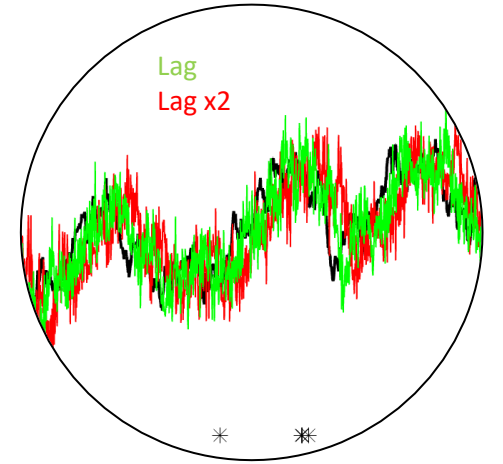
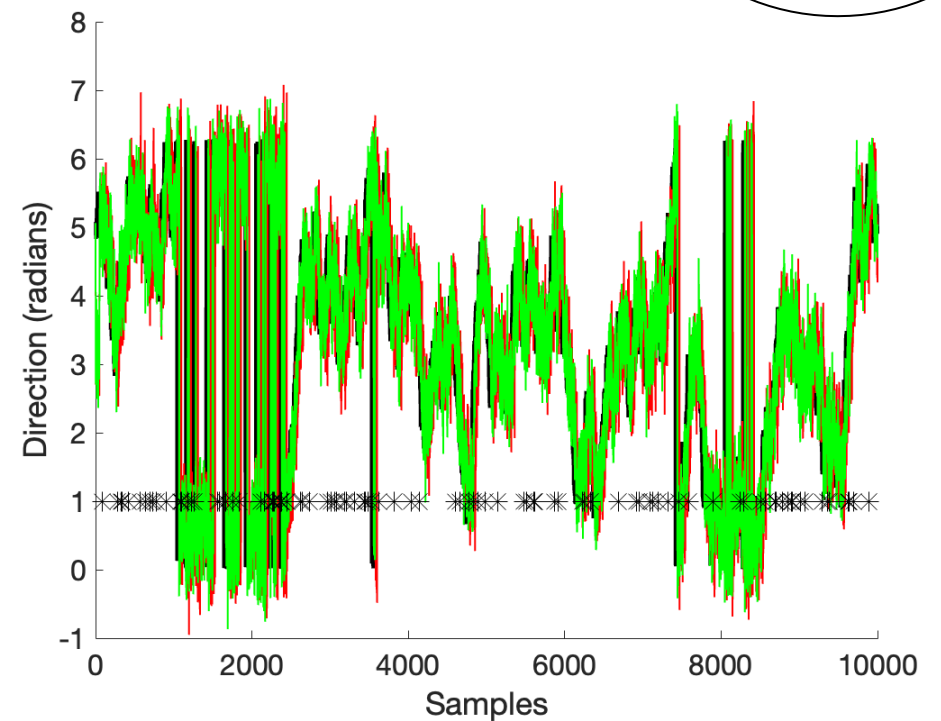
```
if random_subject
    % Random subject direction
    subject1_direction(step)      = pi + nscale*randn;
    subject2_direction(step)      = pi + nscale*randn;
else
    if step > lag
        % Subject direction (noisily) lags behind stimulus direction
        subject1_direction(step)  = stimulus_direction(step-lag) + (nscale*randn);
        subject2_direction(step)  = stimulus_direction(step-lag/2) + (nscale*randn);
    else
        subject1_direction(step)  = pi + nscale*randn;
        subject2_direction(step)  = pi + nscale*randn;
    end
end
```

Example simulation

Random subjects



Tracking subjects



Reward calculation

```
% Probability of reward target appearance per sample
reward_probability = 0.01;

% Check for reward target appearance at sample
if rand < reward_probability
    % Calculate tracking accuracy for each player
    % Assign reward to each player
end
```

Two possible social settings:

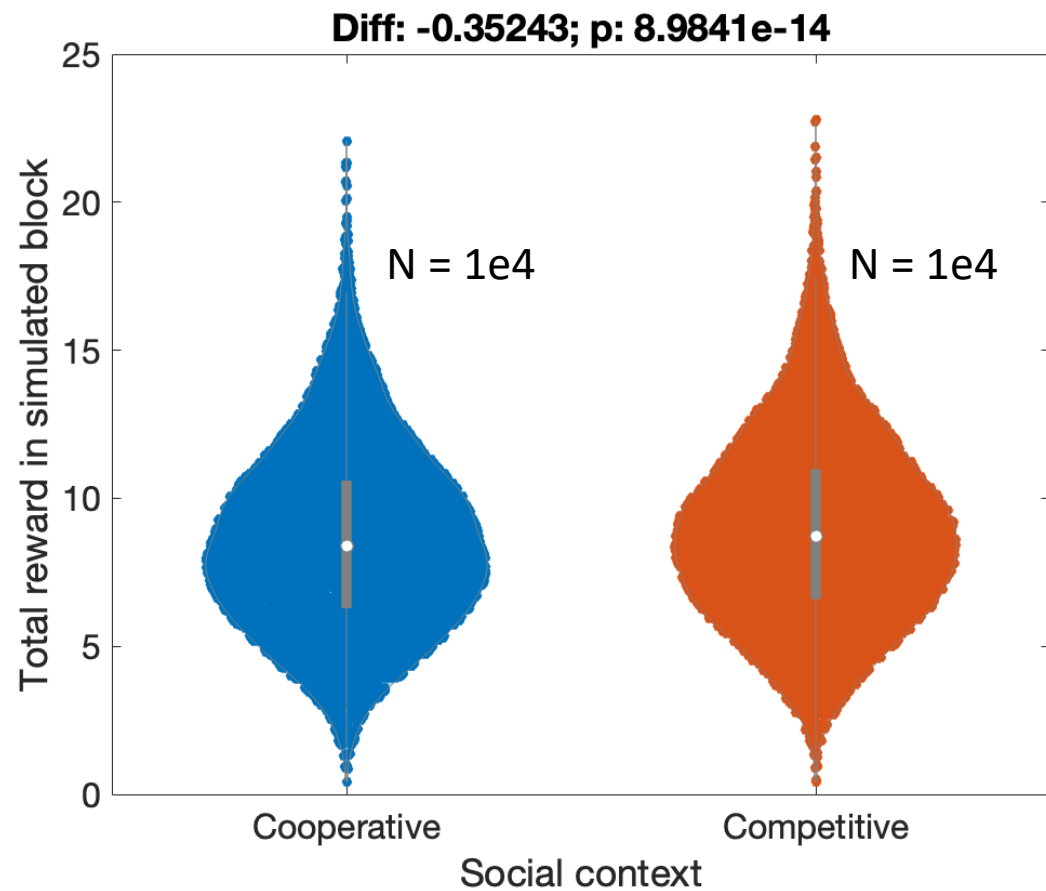
- Option 1: Equally shared reward (cooperative)
- Option 2: Winner- takes-all (competitive)

```
% Determine reward based on accuracy for dyadic hits (both players hit the target)
if accuracy_subject1 > 0.75 && accuracy_subject2 > 0.75 % Corresponds to fixed 0.5 eccentricity
    if strcmp(social_context, 'coop')
        % Cooperative context: Equally shared reward between players
        reward1      = mean([accuracy_subject1 accuracy_subject2]) / 2;
        reward2      = mean([accuracy_subject1 accuracy_subject2]) / 2;
    elseif strcmp(social_context, 'comp')
        % Competitive context: Winner-takes-all if both hit the target
        if accuracy_subject1 > accuracy_subject2
            reward1      = accuracy_subject1;
            reward2      = 0;
        else
            reward1      = 0;
            reward2      = accuracy_subject2;
        end
    end
end
else
    % No reward otherwise
    reward1      = 0;
    reward2      = 0;
end
```

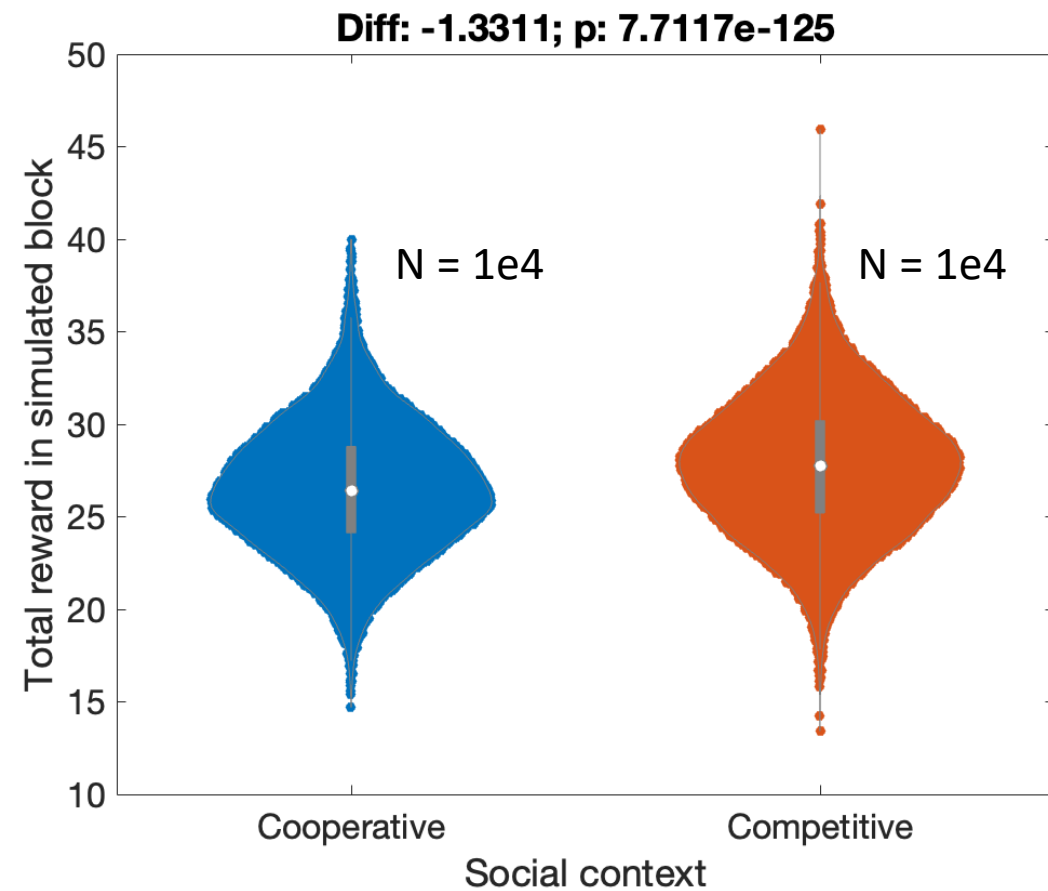
Average reward outcome

- For each simulation ($N = 1e4$ for each social condition):
 - (1) Target-wise reward summed for each player, [...]
 - (2) [...] then averaged across both players.
- Statistics (between cooperative and competitive context):
 - Median difference of average reward per simulation
 - Paired Wilcoxon-signed rank test

Random subject



Tracking subject



Small but significant difference between social condition for random and tracking players