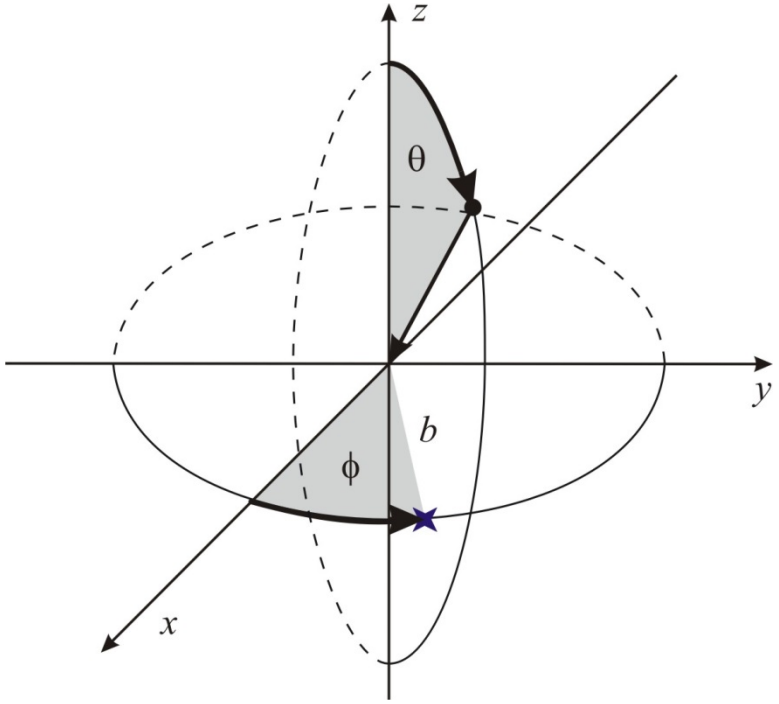
Probability Estimation Case Study: Molecular Dynamics

24 Carbon Vapor Deposition (CVD)

- A carbon dimer is projected toward a diamond substrate.
- We will assume that the dimer can react with the substrate in one of three ways:
  - chemisorption (the atoms in the dimer become bound to the substrate),
  - scattering (the atoms bounce off the substrate),
  - desorption (the atoms become bound to the substrate for a period of time, but are then released).

Variable Definitions

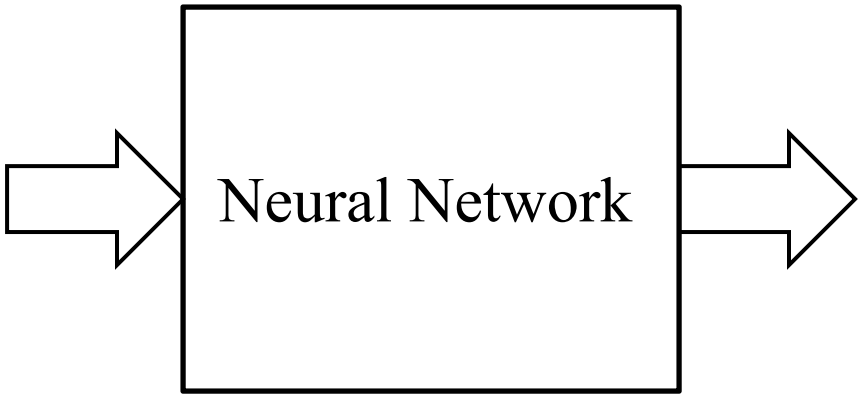
- Black circle represents the carbon dimer,
- Directed line represents the direction of the initial velocity vector.
- Blue star represents the location of the central carbon atom in the diamond substrate.
- Angle θ denotes the angle of incidence, i.e., the angle between the direction of the initial velocity vector of the carbon dimer and the perpendicular on the surface (the *z* direction).



- Impact parameter *b* is the distance between the location of the central atom and the point of intersection of the initial velocity vector and the diamond surface (the origin of the axes).
- Angle ϕ represents the angle between the *x* axis and the line from the origin to the central atom.

Neural Network

$$\mathbf{p} = \begin{bmatrix} \theta \\ \varphi \\ b \\ \nu_{trans} \\ \nu_{rot} \end{bmatrix}$$



$$\mathbf{a}(\mathbf{p}) = \begin{bmatrix} P_C^{NN}(\mathbf{p}) \\ P_S^{NN}(\mathbf{p}) \\ P_C^{NN}(\mathbf{p}) \end{bmatrix}$$

ν_{trans} - translational velocity of the C_2 dimer

ν_{rot} - rotational velocity of the C_2 dimer

$P_C^{NN}(\mathbf{p})$ - NN prediction of chemisorption probability

$P_S^{NN}(\mathbf{p})$ - NN prediction of scattering probability

$P_D^{NN}(\mathbf{p})$ - NN prediction of desorption probability

24

Monte Carlo Data Collection

- Data for training the neural network are obtained by molecular dynamics (MD) simulations, where the motion of atoms and molecules in a material under a given force are simulated, using known laws of physics to calculate the forces on individual atoms
- We use a total of 324 atoms to model the CVD system. Out of these, 282 atoms of diamond substrate are used to model the crystalline face with 40 atoms of hydrogen on the top layer of the diamond surface, and 2 atoms in the C_2 dimer.
- The term Monte Carlo refers to the set of simulations that are obtained by setting a number of the variables to random values for each trajectory. We refer to the simulation of a single trajectory as an MD simulation, since the principles of molecular dynamics are used to perform the computations.

Training Data

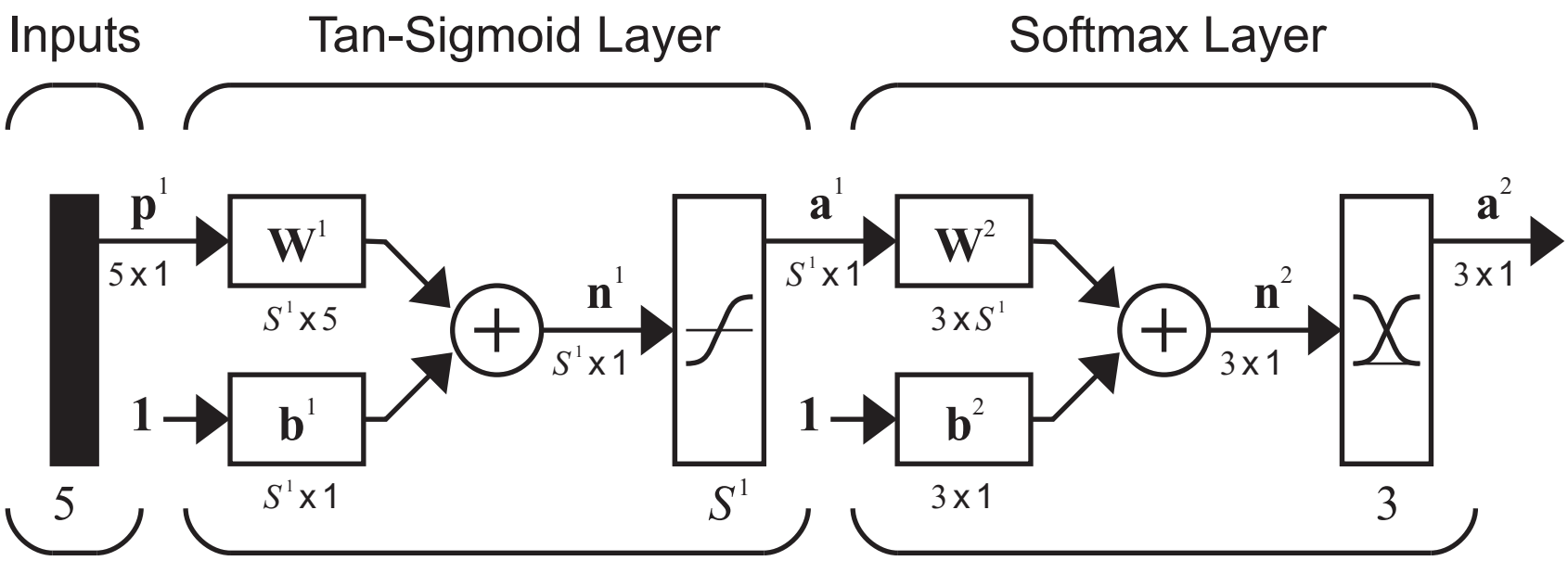
The targets are obtained by estimating the probabilities of chemisorption, scattering or desorption from the Monte Carlo trials:

$$P_X^{MC}(\mathbf{p}) = \frac{N_X}{N_T}$$

where N_X is the number of MD trajectories that resulted in reaction X, and N_T is the total number of MD trajectories computed in the Monte Carlo trials. Since we do not know the true underlying reaction probabilities, we use the estimates obtained from the Monte Carlo trials as target outputs for the neural network. We can think of these estimates as noisy versions of the true probabilities.

2000 different input/target pairs were generated. Of these, 70% were randomly selected for training, 15% for validation, and 15% for testing. For each trajectory, the **p** were generated randomly, using physically-appropriate distributions for each variable. A total of 50 different trajectories were run to obtain each estimated probability. This means that 2000x50 trajectories were run to create the entire data set.

Network Architecture



$$\mathbf{a}^1 = \mathbf{tansig}(\mathbf{W}^1\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{softmax}(\mathbf{W}^2\mathbf{a}^1 + \mathbf{b}^2)$$

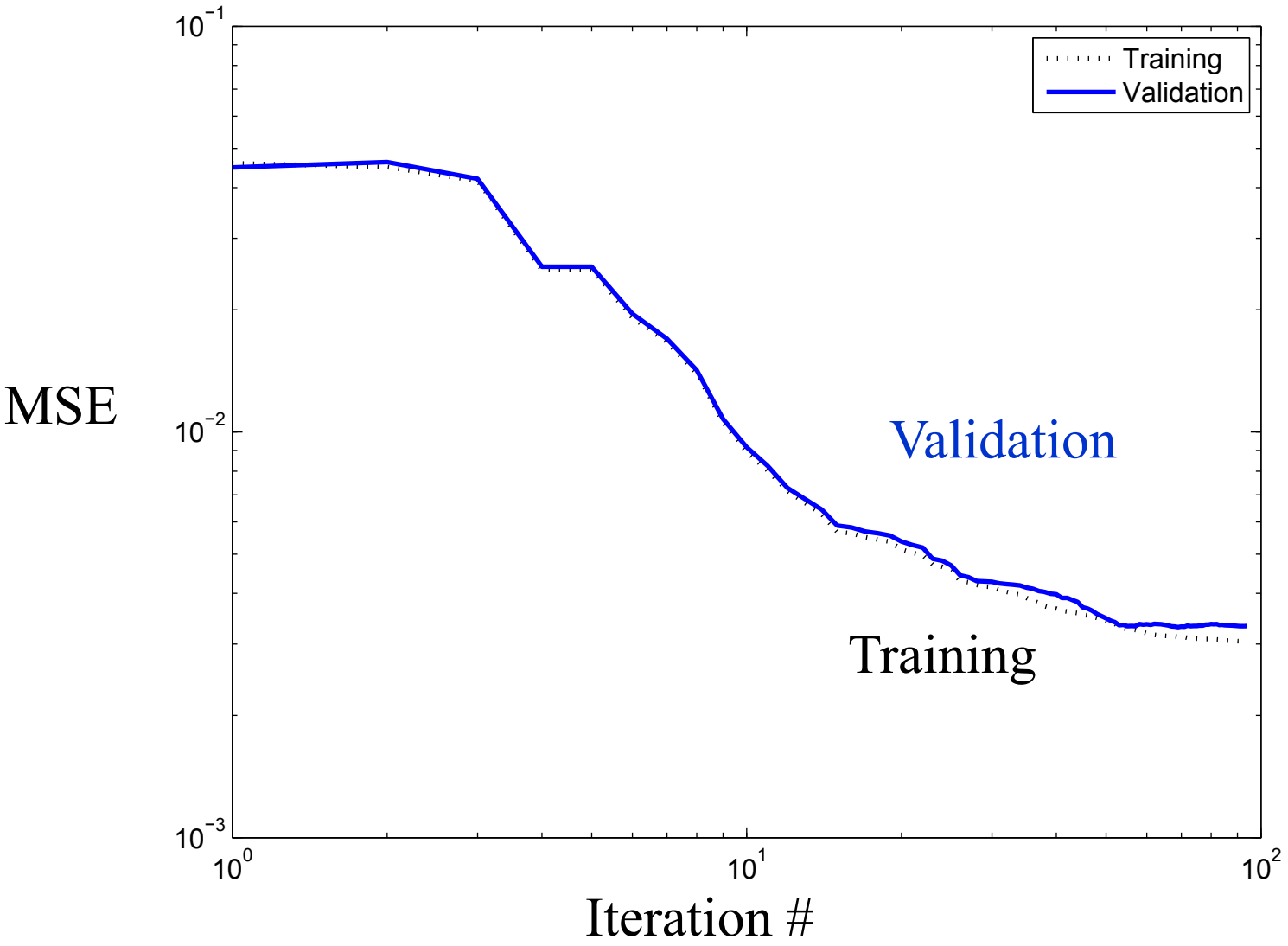
$$\mathbf{t} = \begin{bmatrix} P_C^{MC}(\mathbf{p}) \\ P_S^{MC}(\mathbf{p}) \\ P_C^{MC}(\mathbf{p}) \end{bmatrix}$$

Softmax Transfer Function

$$a_i = f(n_i) = \frac{\exp(n_i)}{\sum_{j=1}^{S} \exp(n_j)}$$

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} a_1^m\left(\sum_{i=1}^{S^m} a_i^m - a_1^m\right) & -a_1^m a_2^m & \cdots & -a_1^m a_{S^m}^m \\ -a_2^m a_1^m & a_2^m\left(\sum_{i=1}^{S^m} a_i^m - a_2^m\right) & \cdots & -a_2^m a_{S^m}^m \\ \vdots & \vdots & \ddots & \vdots \\ -a_{S^m}^m a_1^m & -a_{S^m}^m a_2^m & \cdots & a_{S^m}^m\left(\sum_{i=1}^{S^m} a_i^m - a_{S^m}^m\right) \end{bmatrix}$$

Training Performance (S^1=10)



24 Training and Validation Performance

$S^1 = 10$

	Training RMSE	Validation RMSE
$P_C(\mathbf{p})$	0.0496	0.0493
$P_S(\mathbf{p})$	0.0634	0.0659
$P_D(\mathbf{p})$	0.0586	0.0604

$S^1 = 2$

	Training RMSE	Validation RMSE
$P_C(\mathbf{p})$	0.0634	0.0627
$P_S(\mathbf{p})$	0.0669	0.0704
$P_D(\mathbf{p})$	0.0617	0.0618

$S^1 = 20$

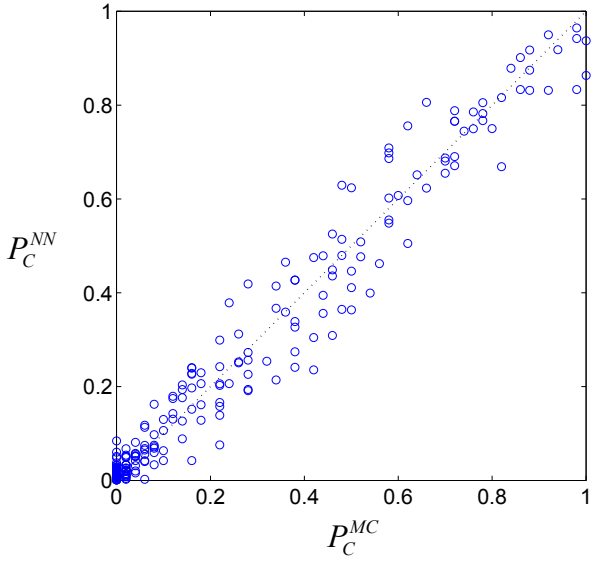
	Training RMSE	Validation RMSE
$P_C(\mathbf{p})$	0.0432	0.0444
$P_S(\mathbf{p})$	0.0603	0.0643
$P_D(\mathbf{p})$	0.0569	0.0595

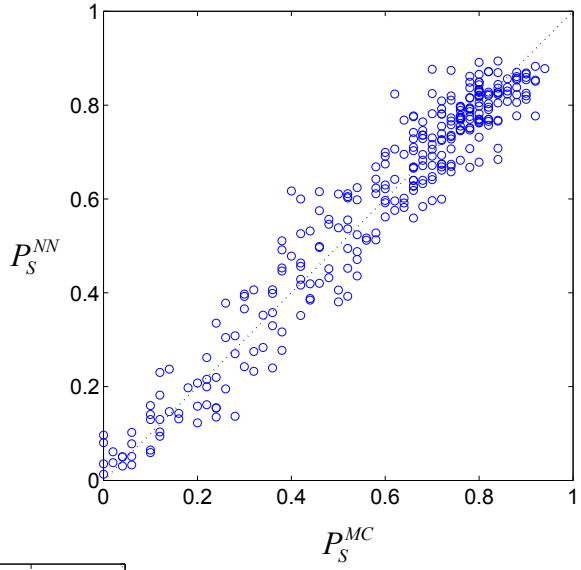
Multiple Initial Conditions

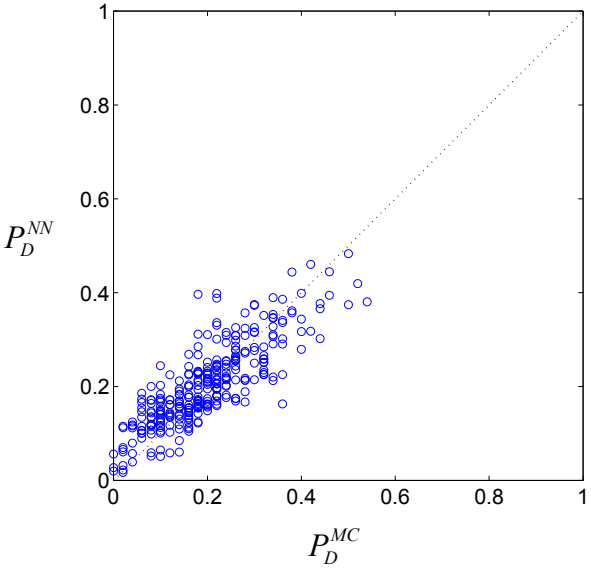
3.046e-003	2.953e-003	3.031e-003	3.105e-003	3.050e-003

- Final validation MSE for five different training runs.
- All of the errors are similar, so we have reached a global minimum at each run.
- If one error was significantly lower than the others, then we would use the weights that obtained the lowest error.

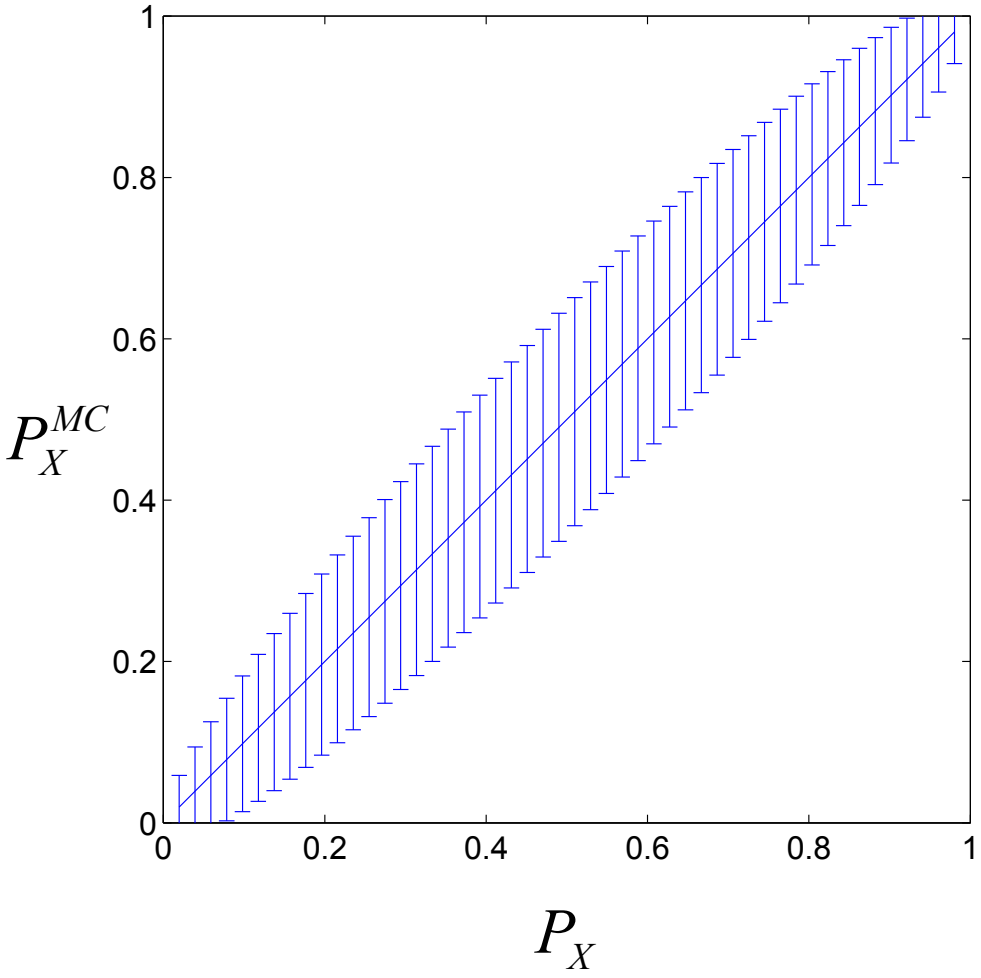
Output vs Target Scatter Plots







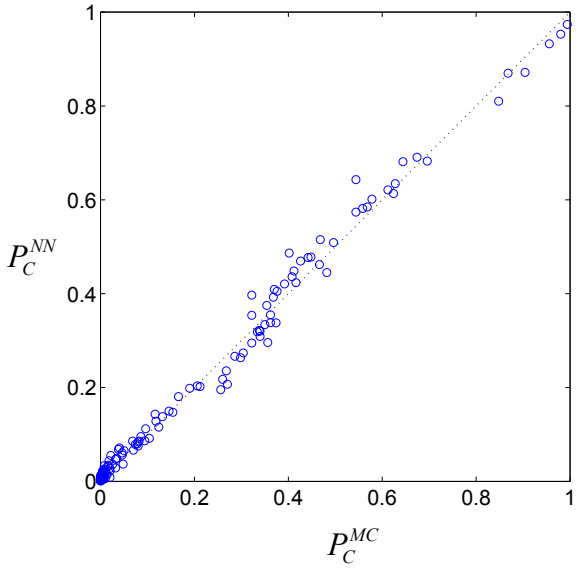
Statistical Error in $P^{MC}{}_X$

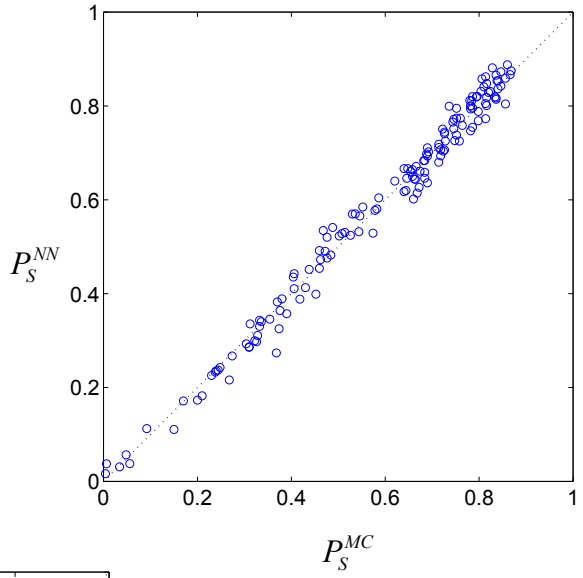


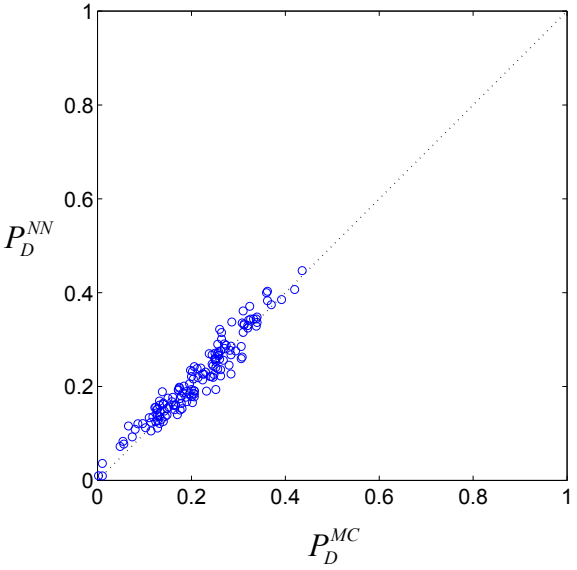
$$P_X - 2\Delta \le P_X^{MC} \le P_X + 2\Delta$$

$$\Delta = \sqrt{\frac{P_X\left(1-P_X\right)}{N_T}}$$

Outputs vs Targets for N_T=500







24 Reaction Prob. vs Impact Parameter