

Using CNN Deep Learning to Find Slow Pions in the PXD

Johannes Bilk, 10. November 2021

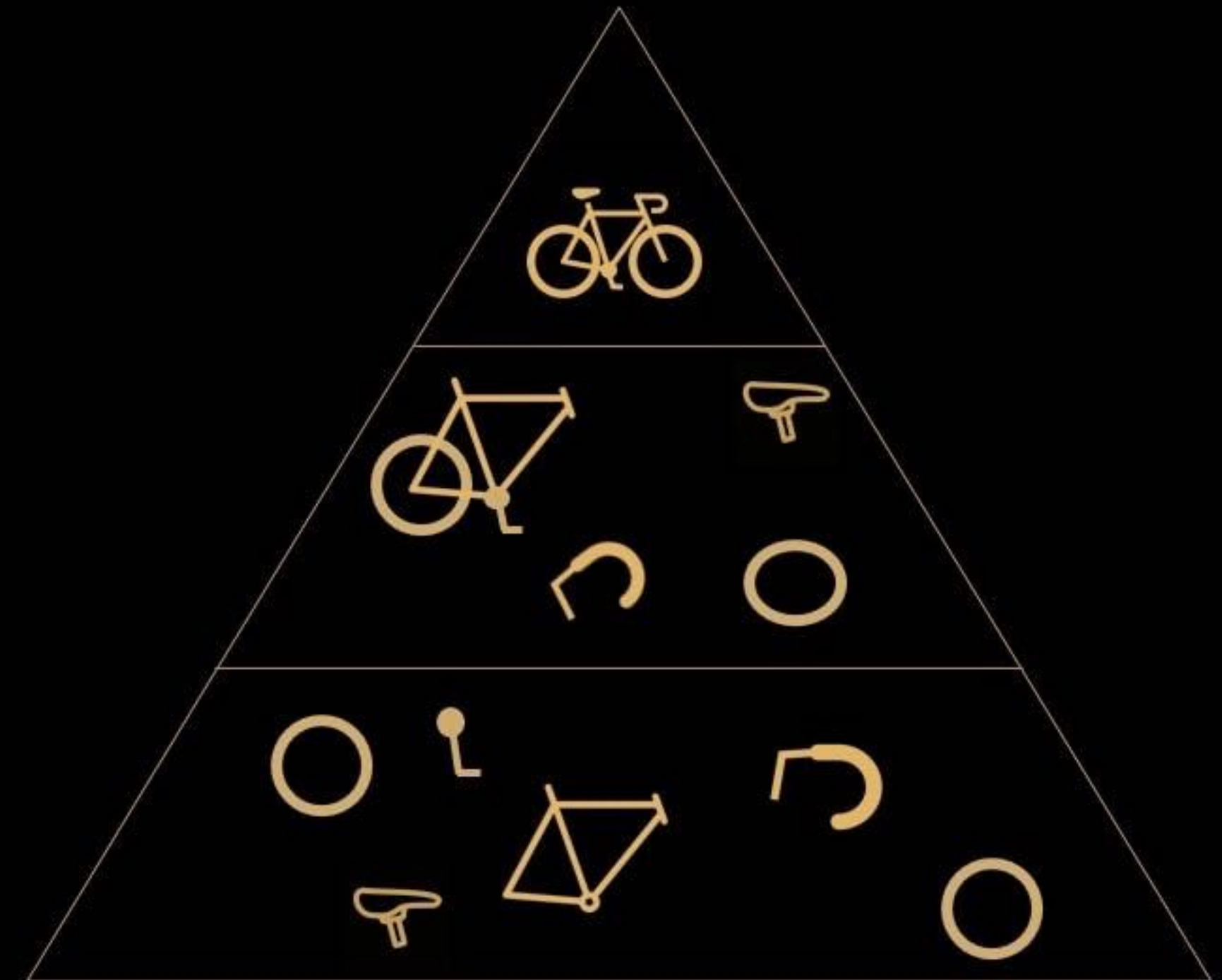
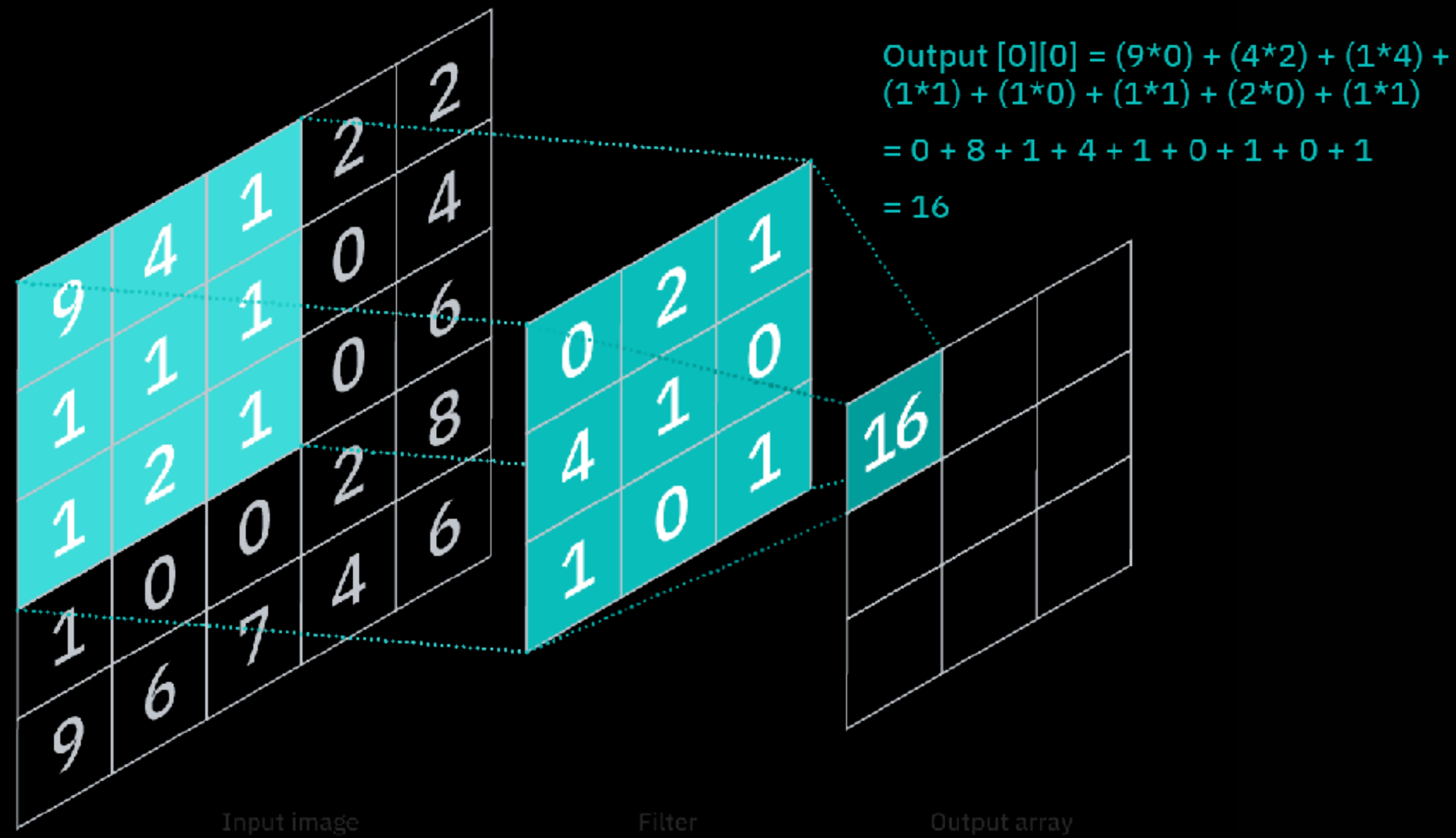
Introduction

Table of Contents

- Introduction
- Prior Findings
- New Results
- Preliminary Results
- Summary & Outlook

Convolutional Layer

What does a convolution do?



Confusion Matrix

The multi-class case

- We have c_n classes
- On the vertical are actual classes
- On the horizontal are predicted classes
- TP, TN, FN and FP can be read directly from the matrix

	$c_0 \dots c_{k-1}$	c_k	$c_{k+1} \dots c_n$
Actual Class $c_{k+1} \dots c_n$	TN	FP	TN
c_k	FN	TP	FN
$c_0 \dots c_{k-1}$	TN	FP	TN

Predicted Class

Prior Findings

The Network Setups

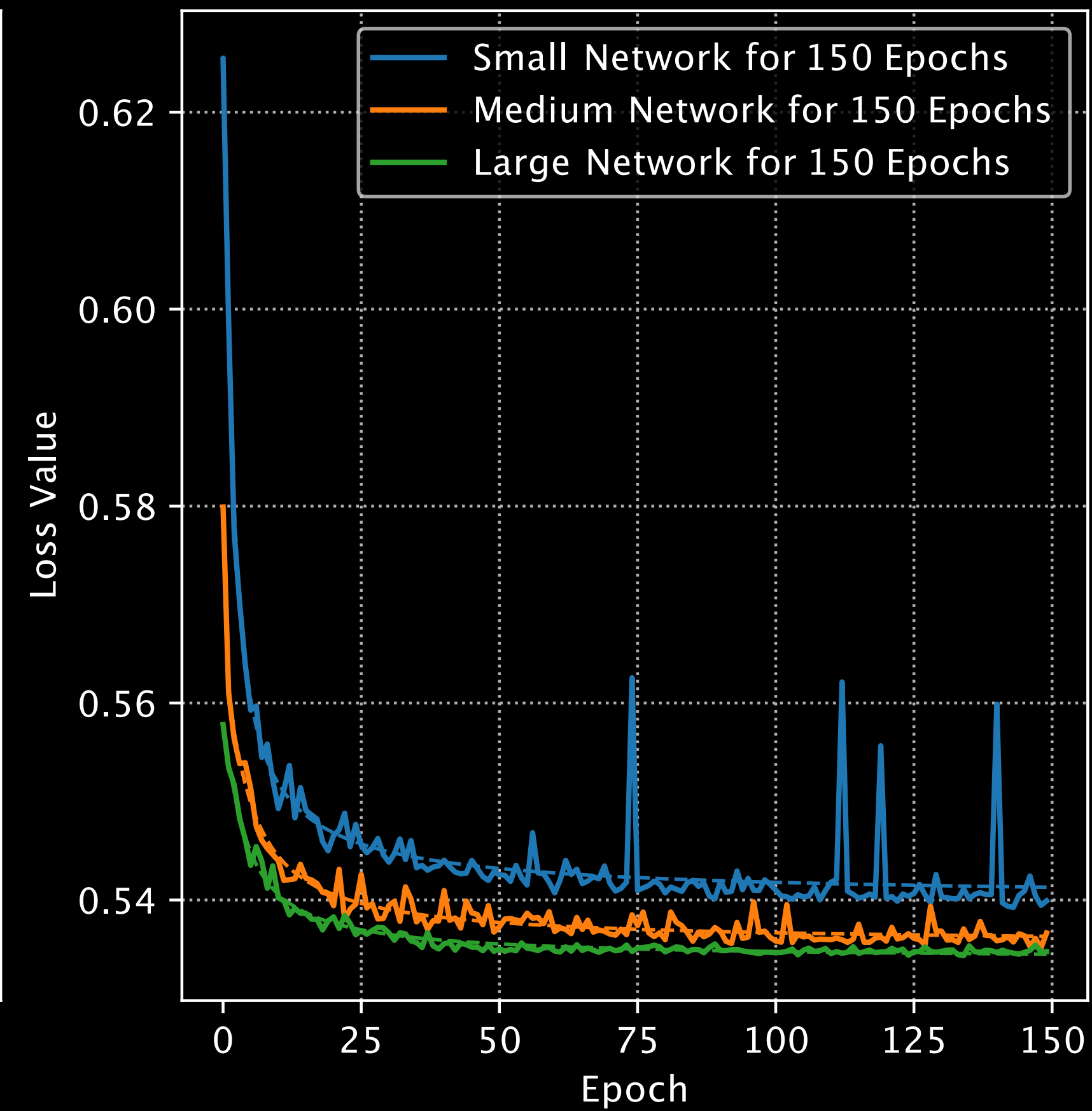
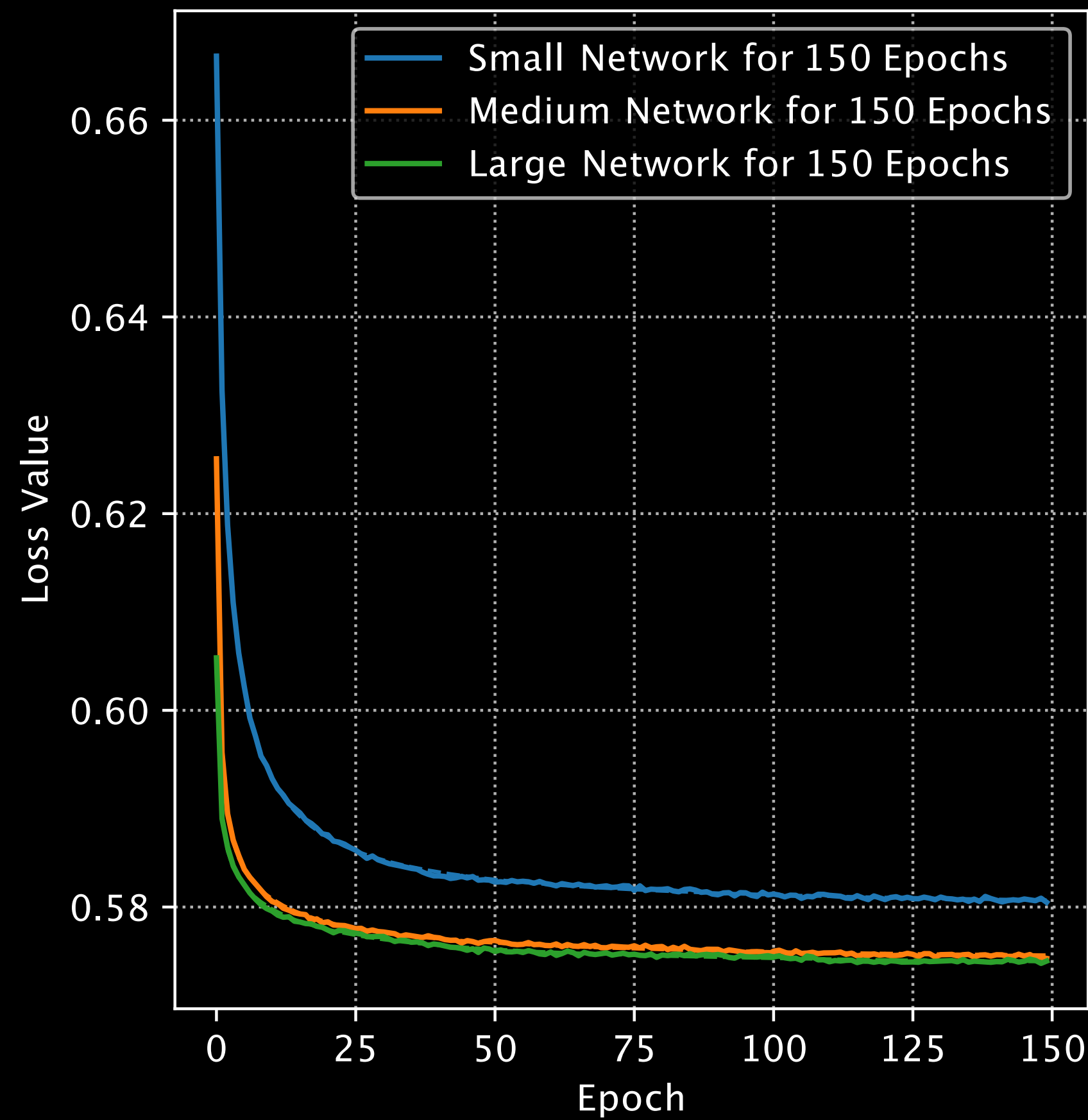
Prior Findings

	<i>Small Network</i>	<i>Medium Network</i>	<i>Large Network</i>
<i>Convolutional Layer</i>	0	1	3
<i>Output Channels</i>	-	3	5
<i>Kernel Size</i>	-	3	3
<i>Padding</i>	-	1	1
<i>Activation Function</i>	-	ReLU	ReLU
<i>Linear Layer</i>	5	5	5
<i>Layer Width</i>	81	243	405
<i>Dropout Rate</i>	50 %	50 %	50 %
<i>Activation Function</i>	ReLU	ReLU	ReLU

Comparing the Setups

Prior Findings

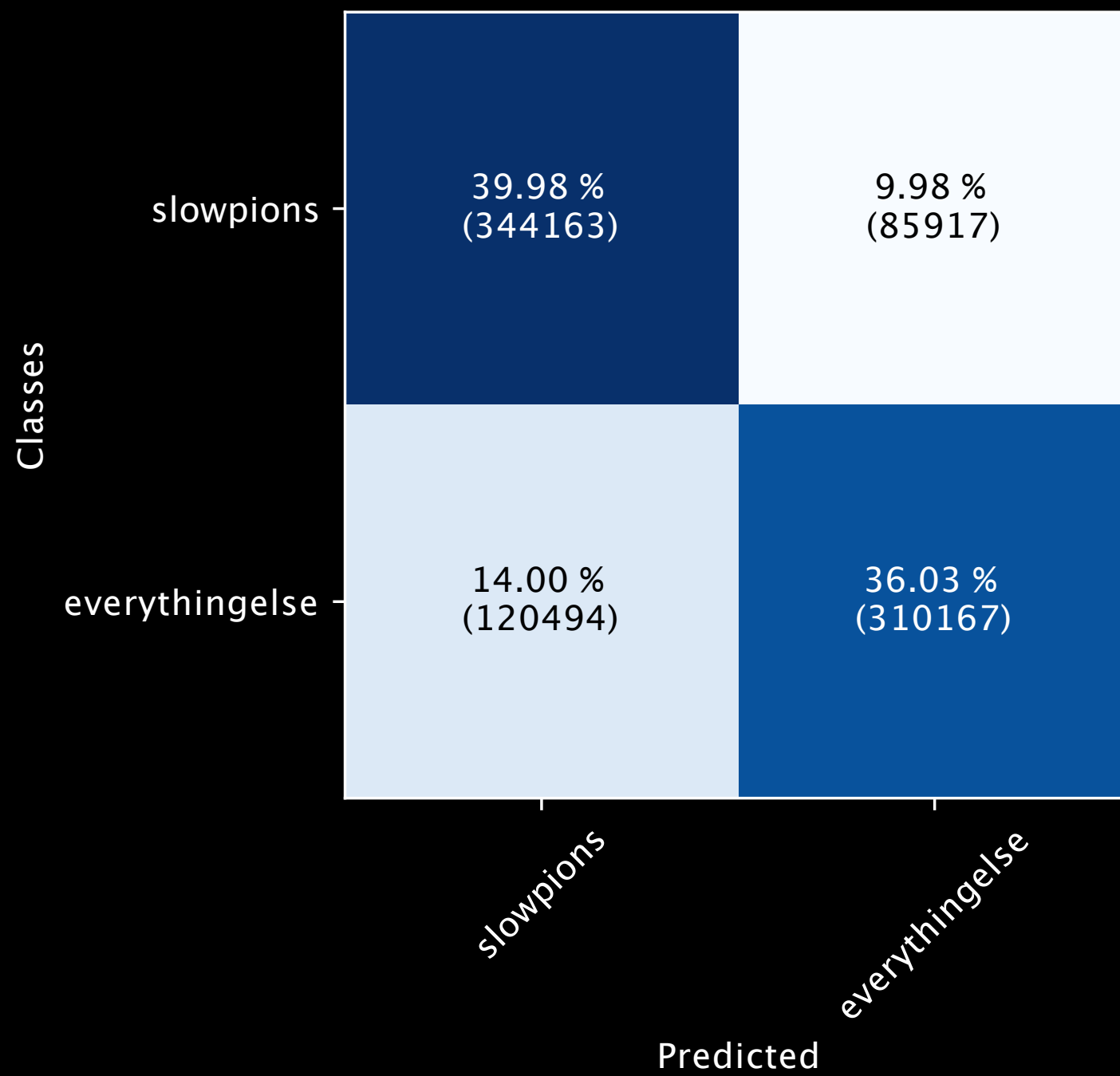
Loss Curves



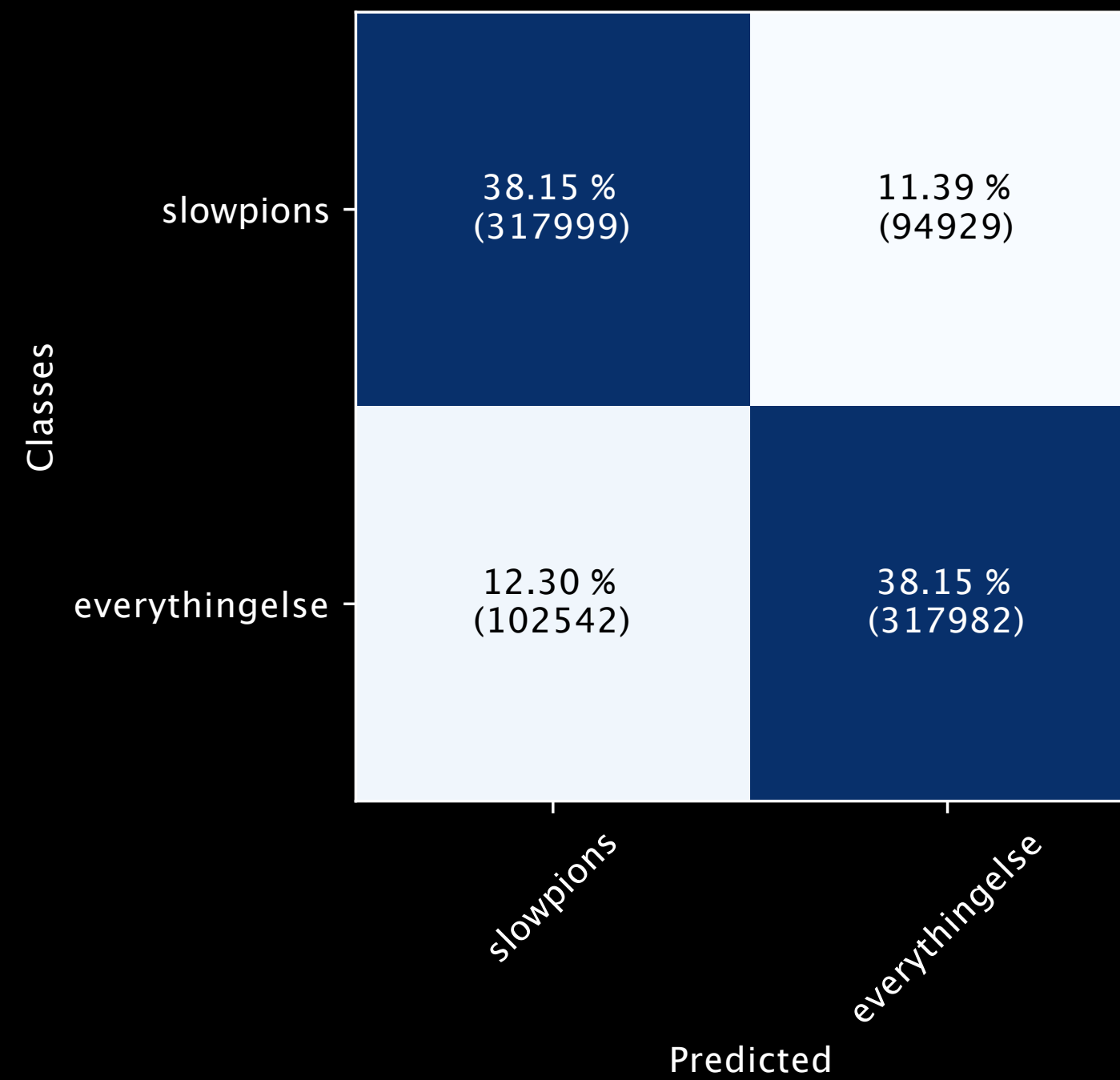
Comparing the Setups

Prior Findings

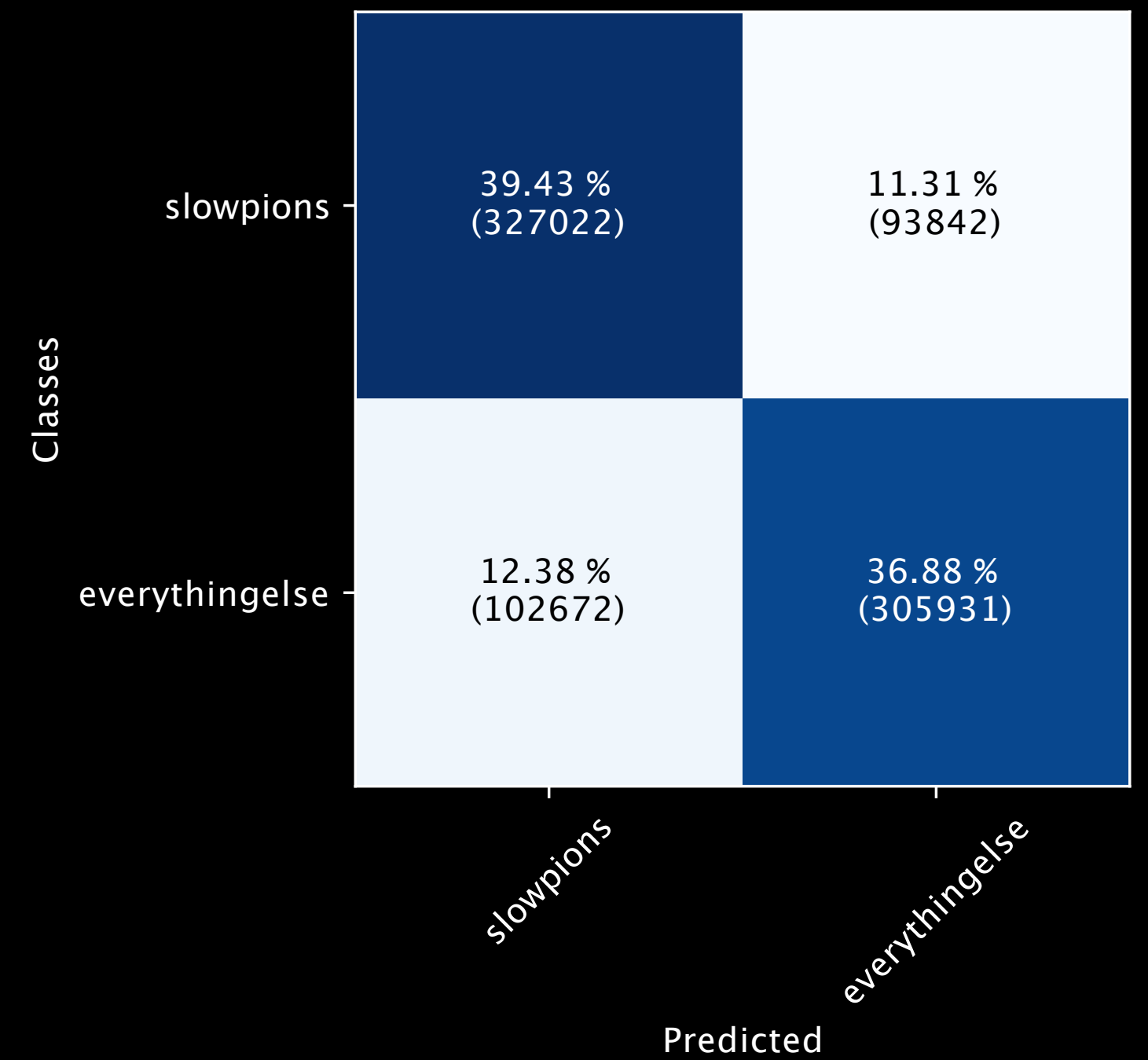
Confusion Matrix for Small Network for 25 Epochs



Confusion Matrix for Medium Network for 25 Epochs



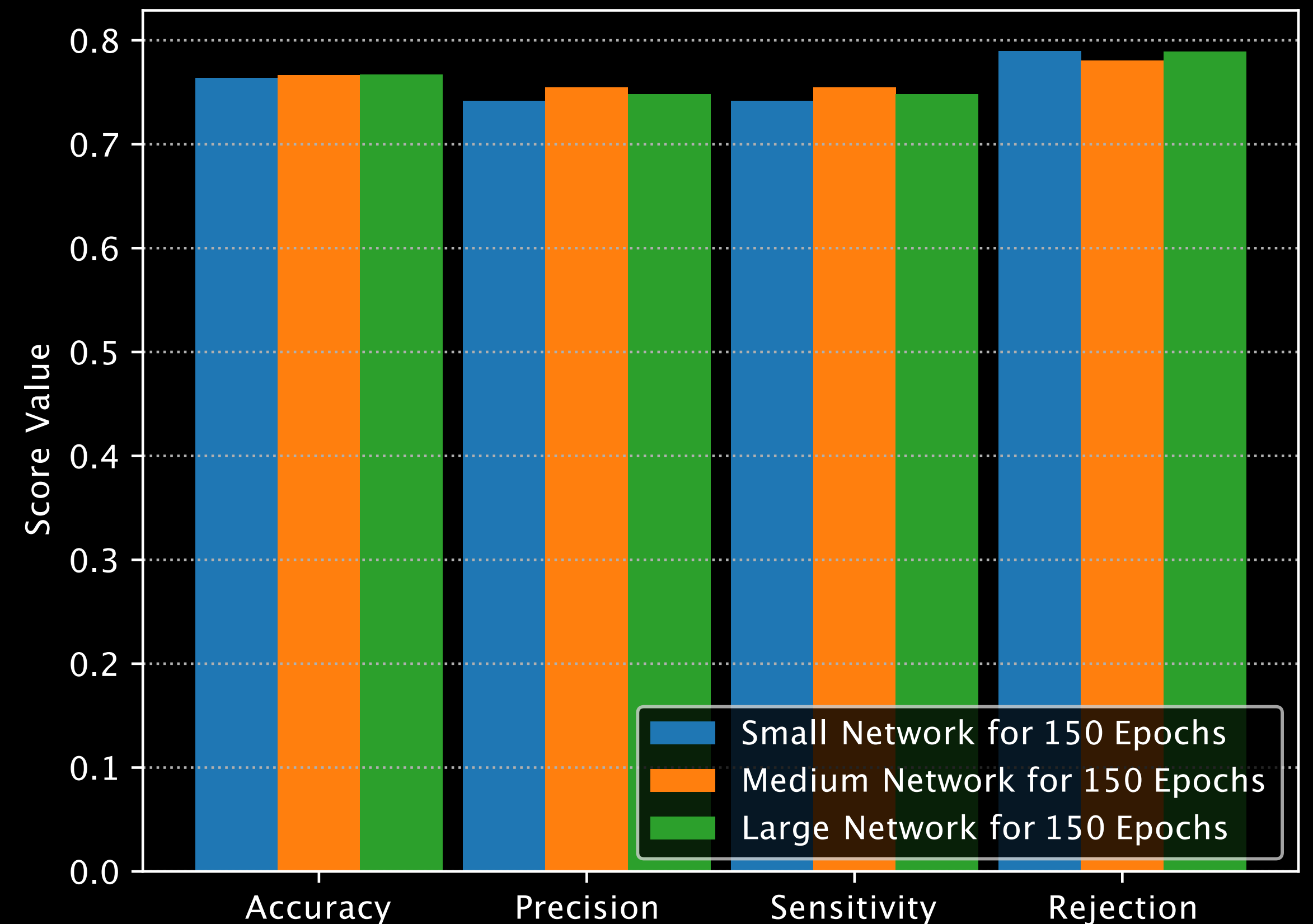
Confusion Matrix for Large Network for 25 Epochs



Comparing the Setups

Prior Findings

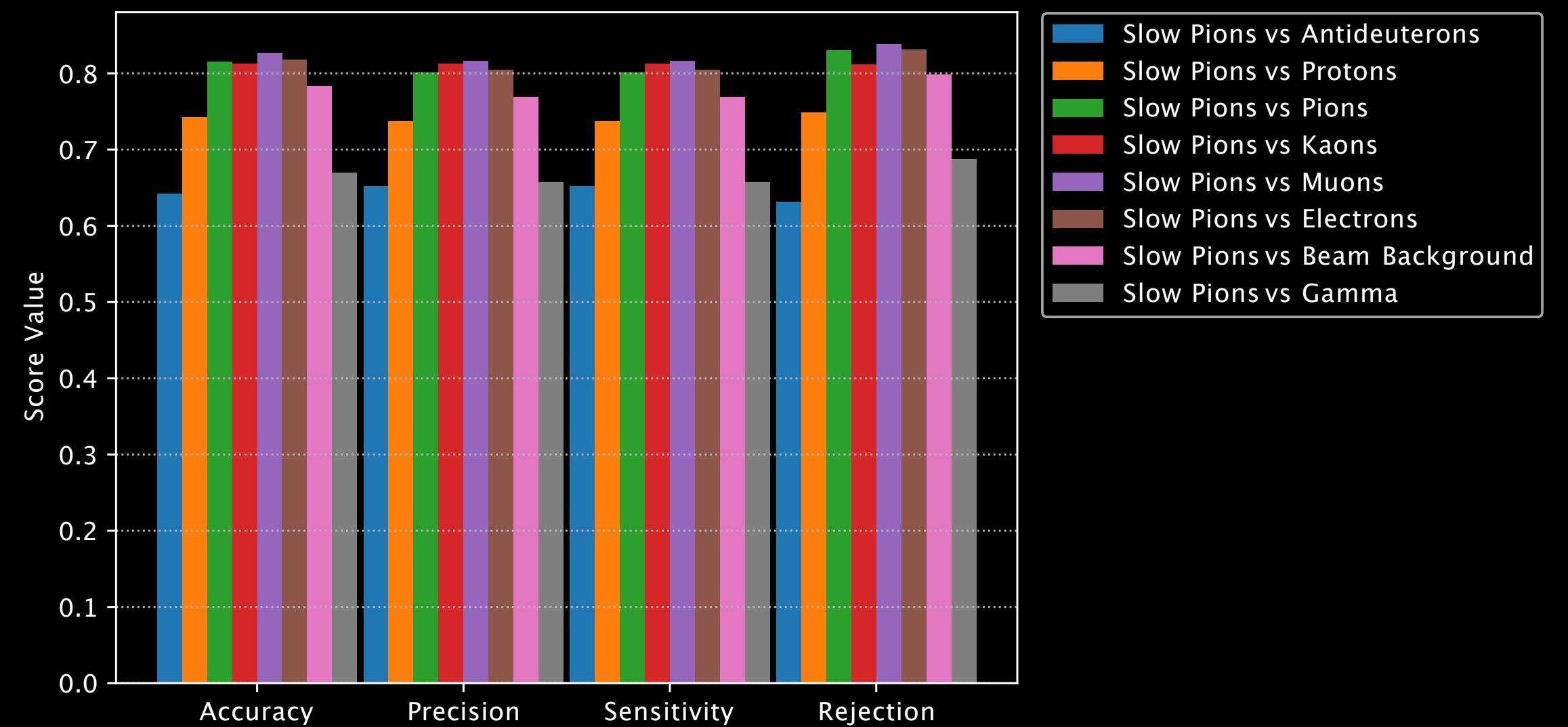
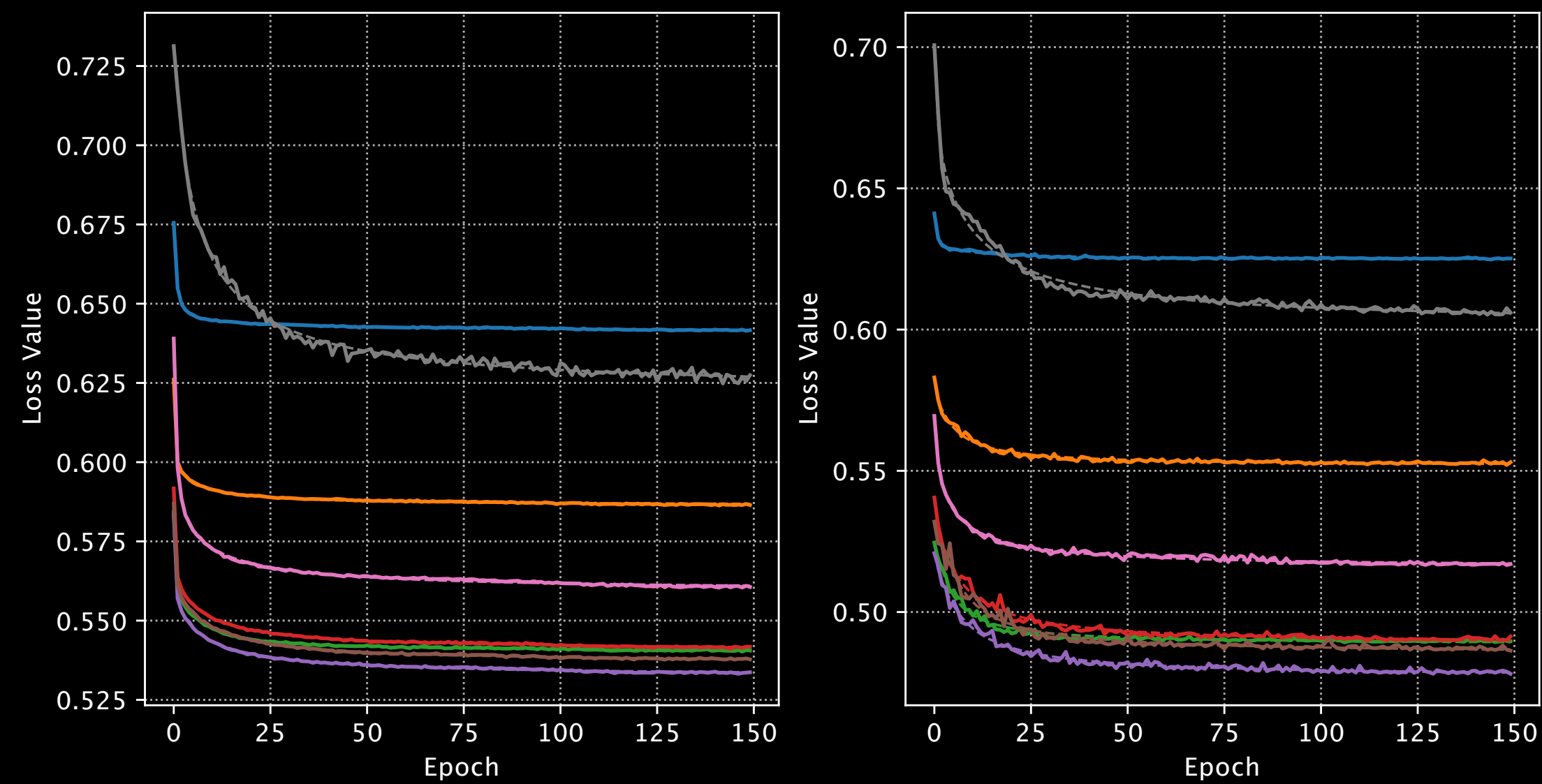
- performance of all three very similar
- difference in scores potentially due to statistical noise
- the loss curves for large NN showed better generalisation



Tests against Individual Categories

Prior Findings

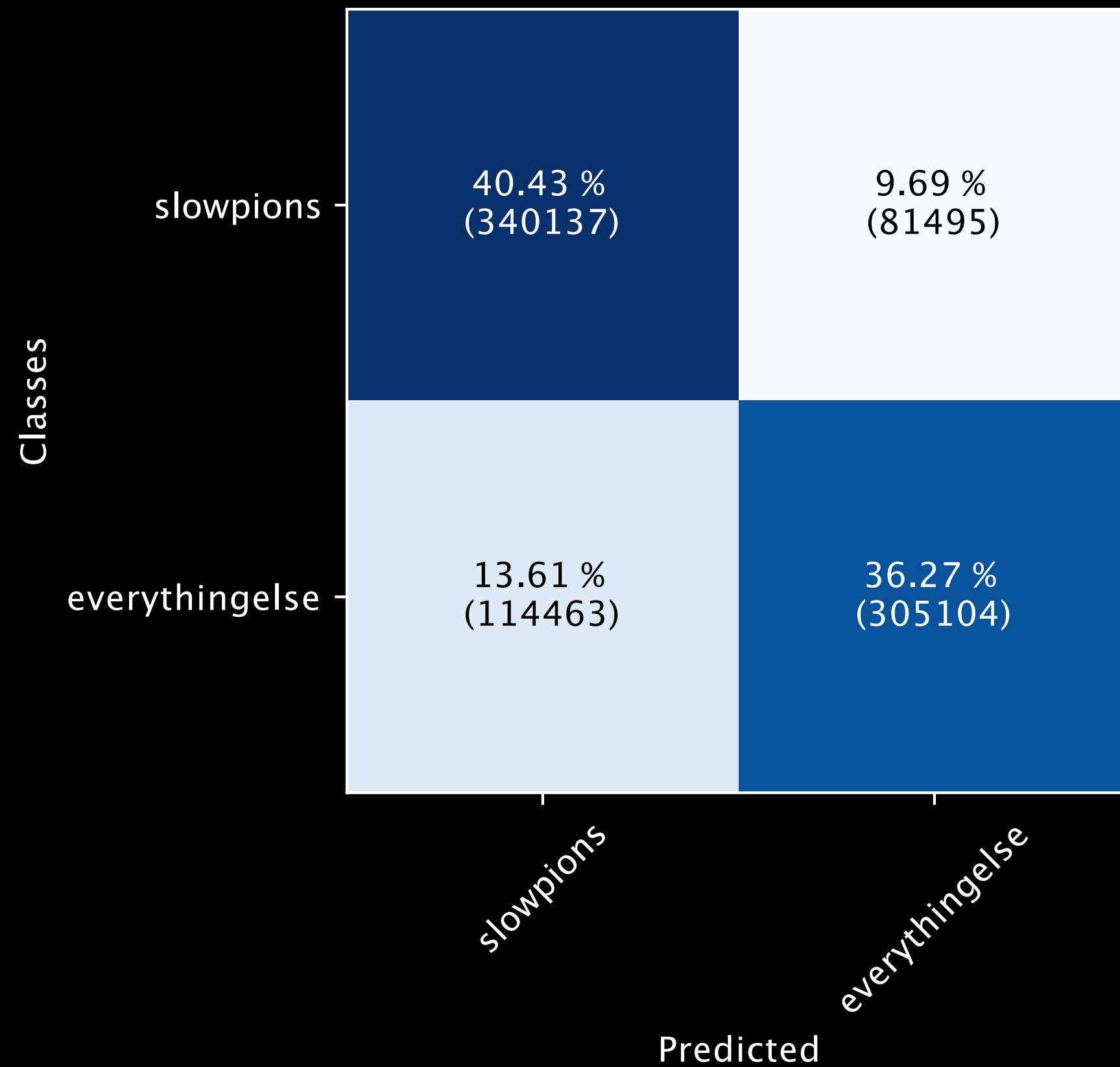
Loss Curves



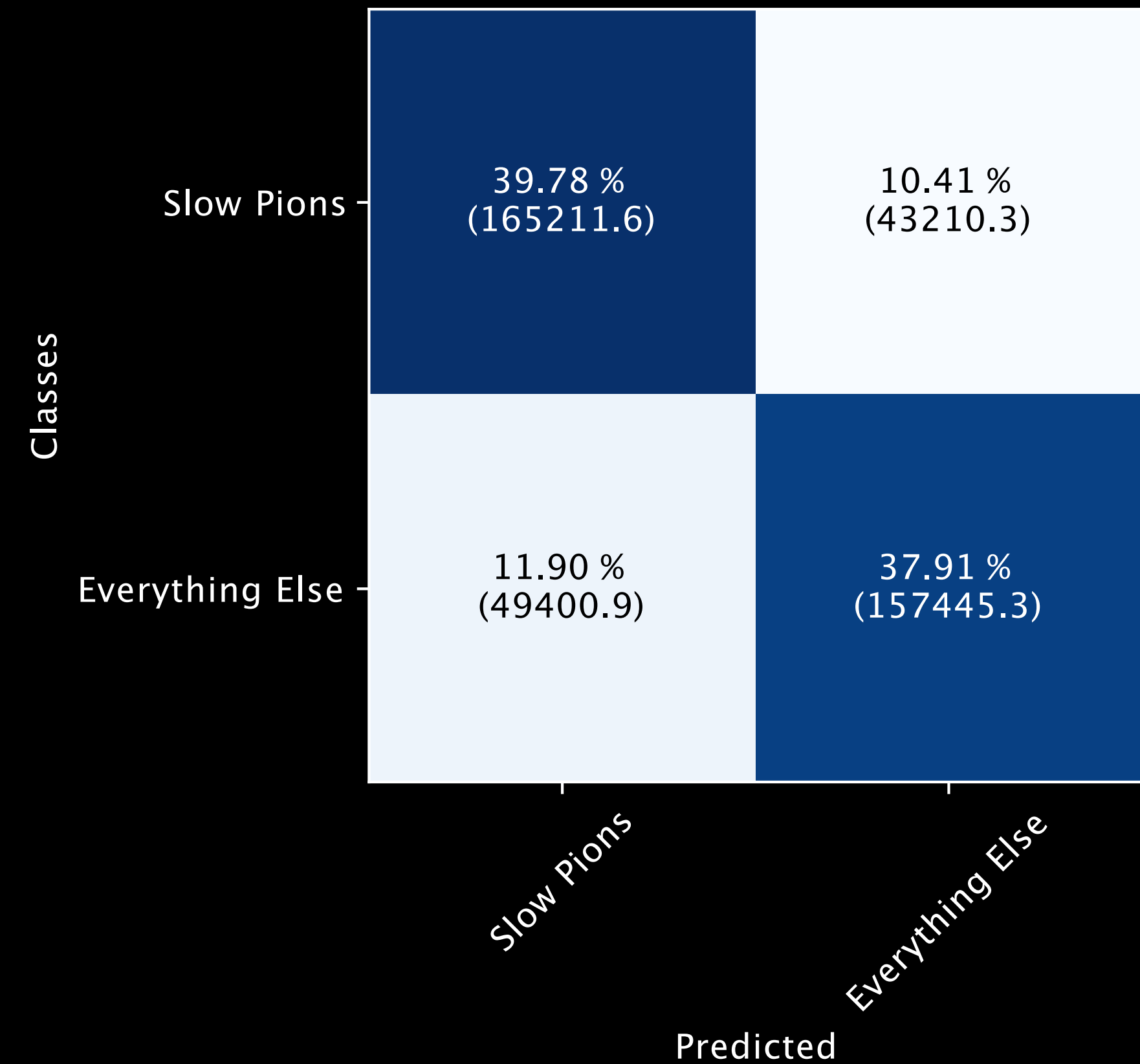
Everything at once against Slow Pions

Prior Findings

Confusion Matrix for Binary Case



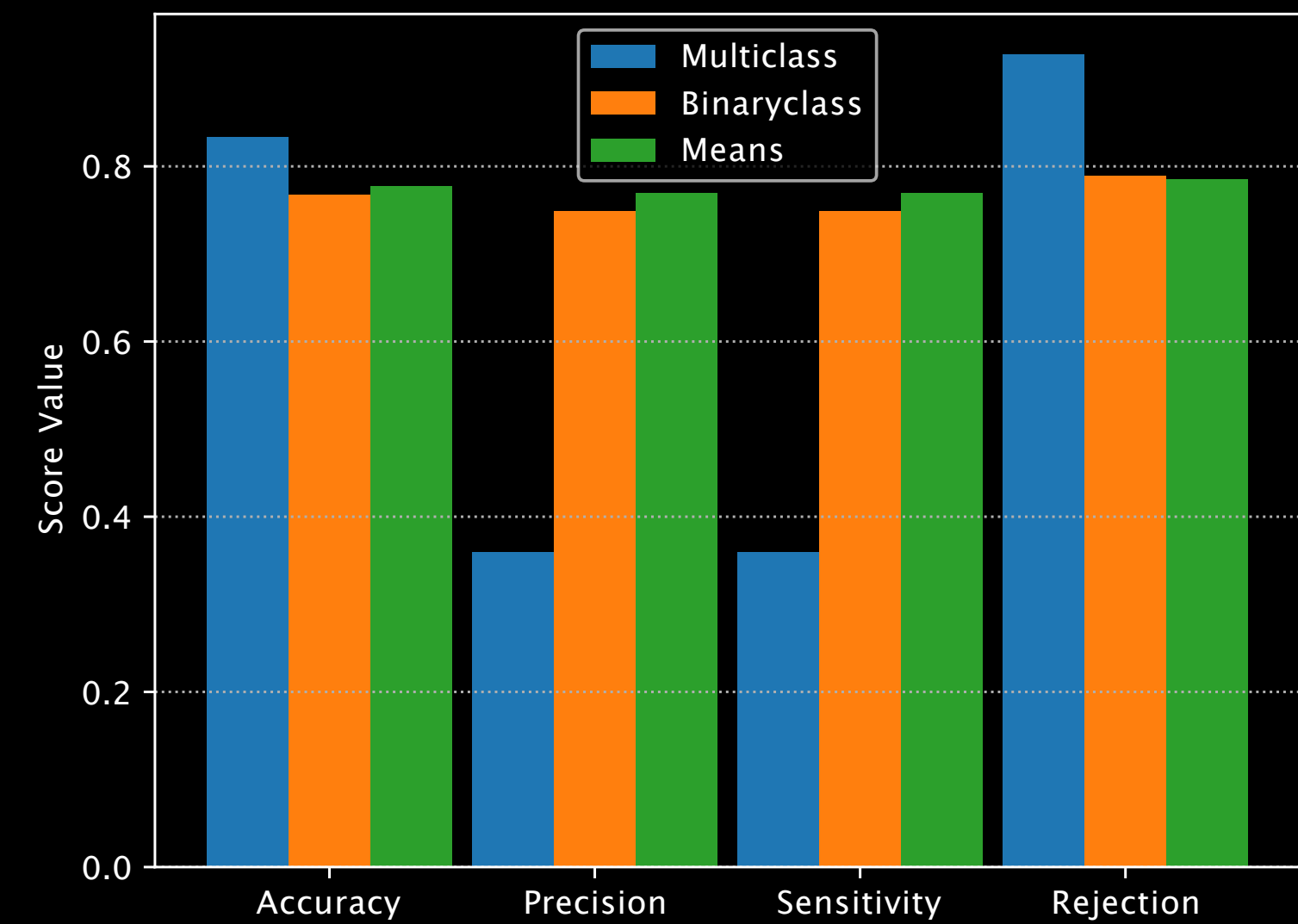
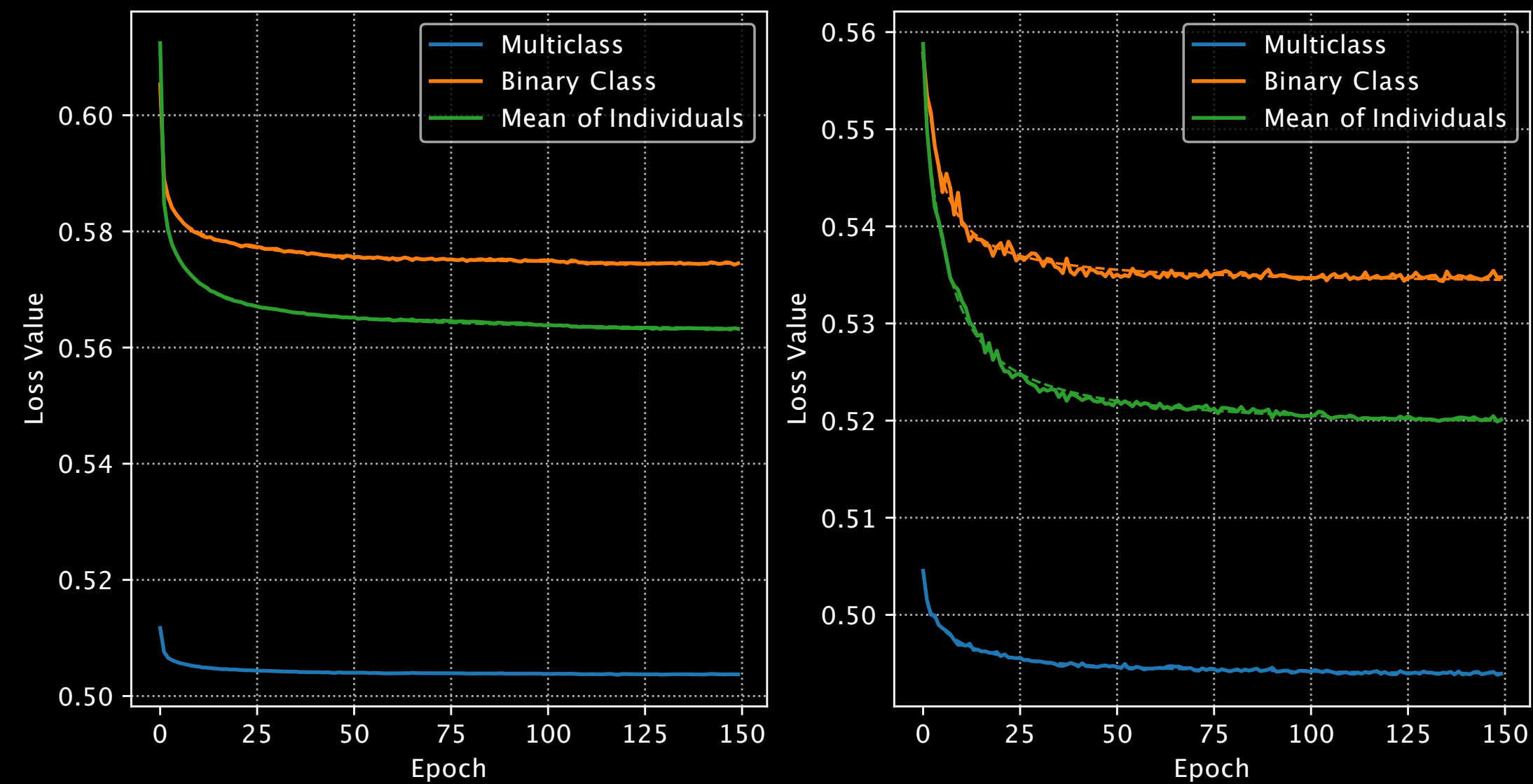
Confusion Matrix for Mean of Individuals



Everything at once against Slow Pions

Prior Findings

Loss Curves



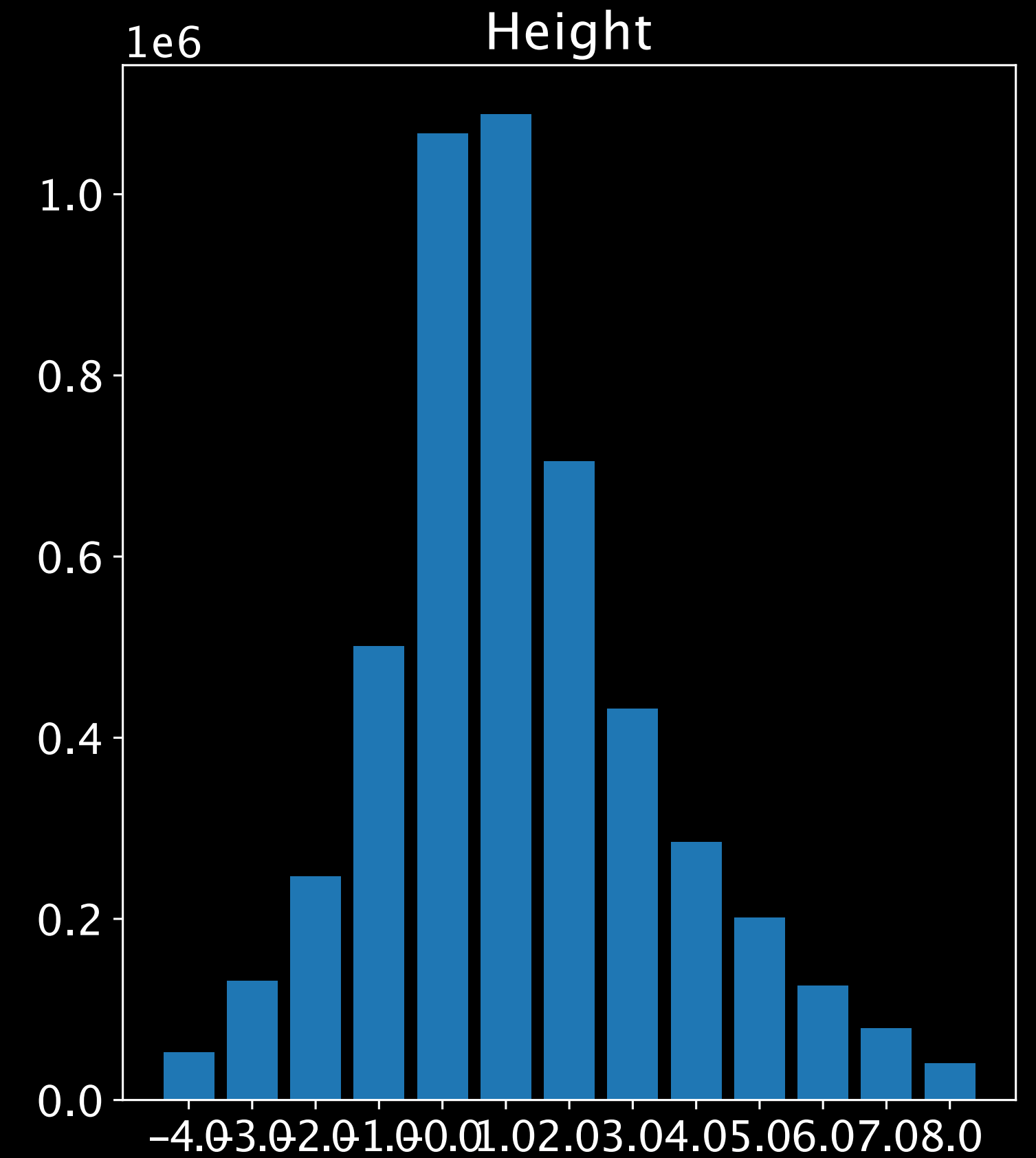
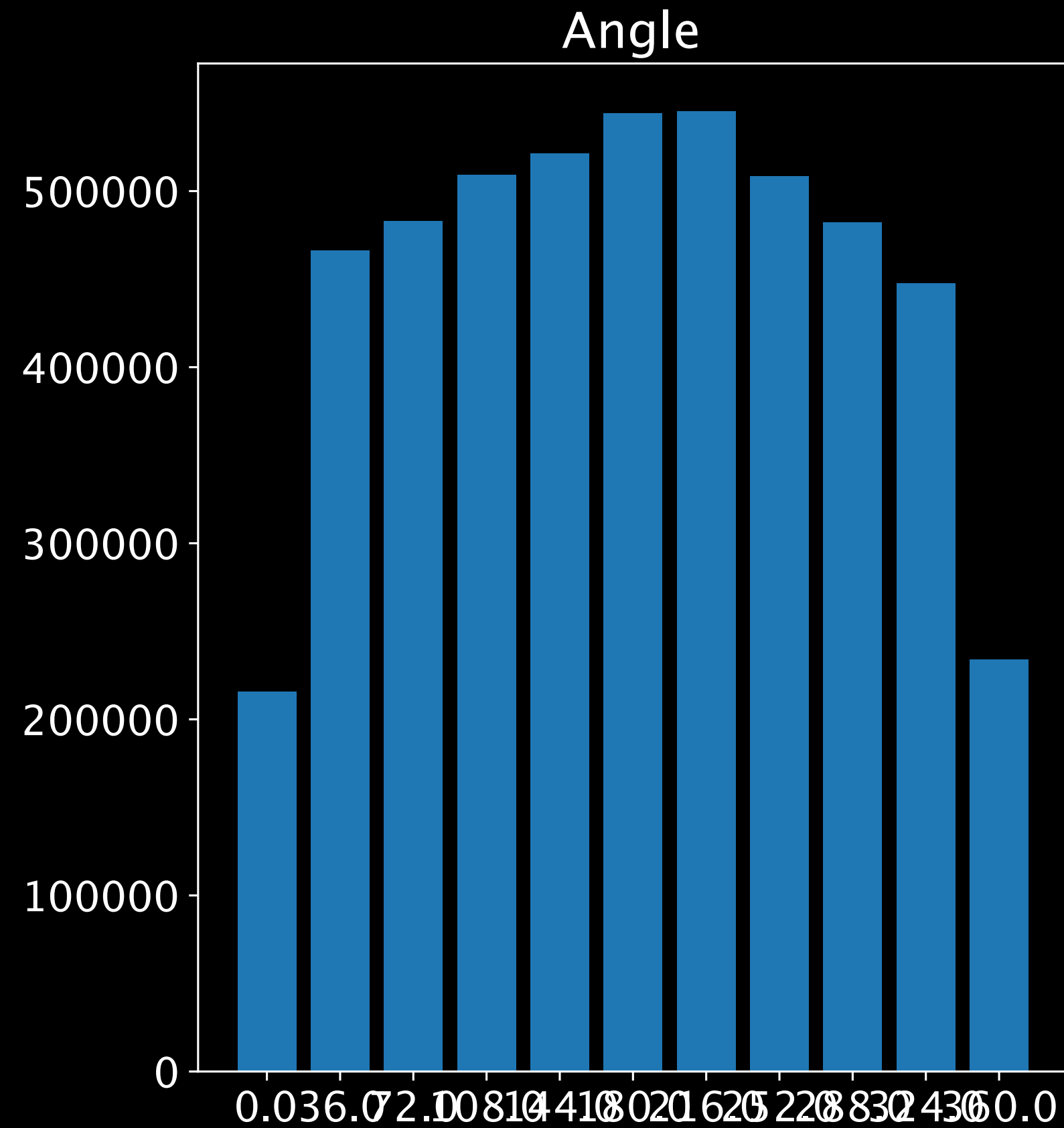
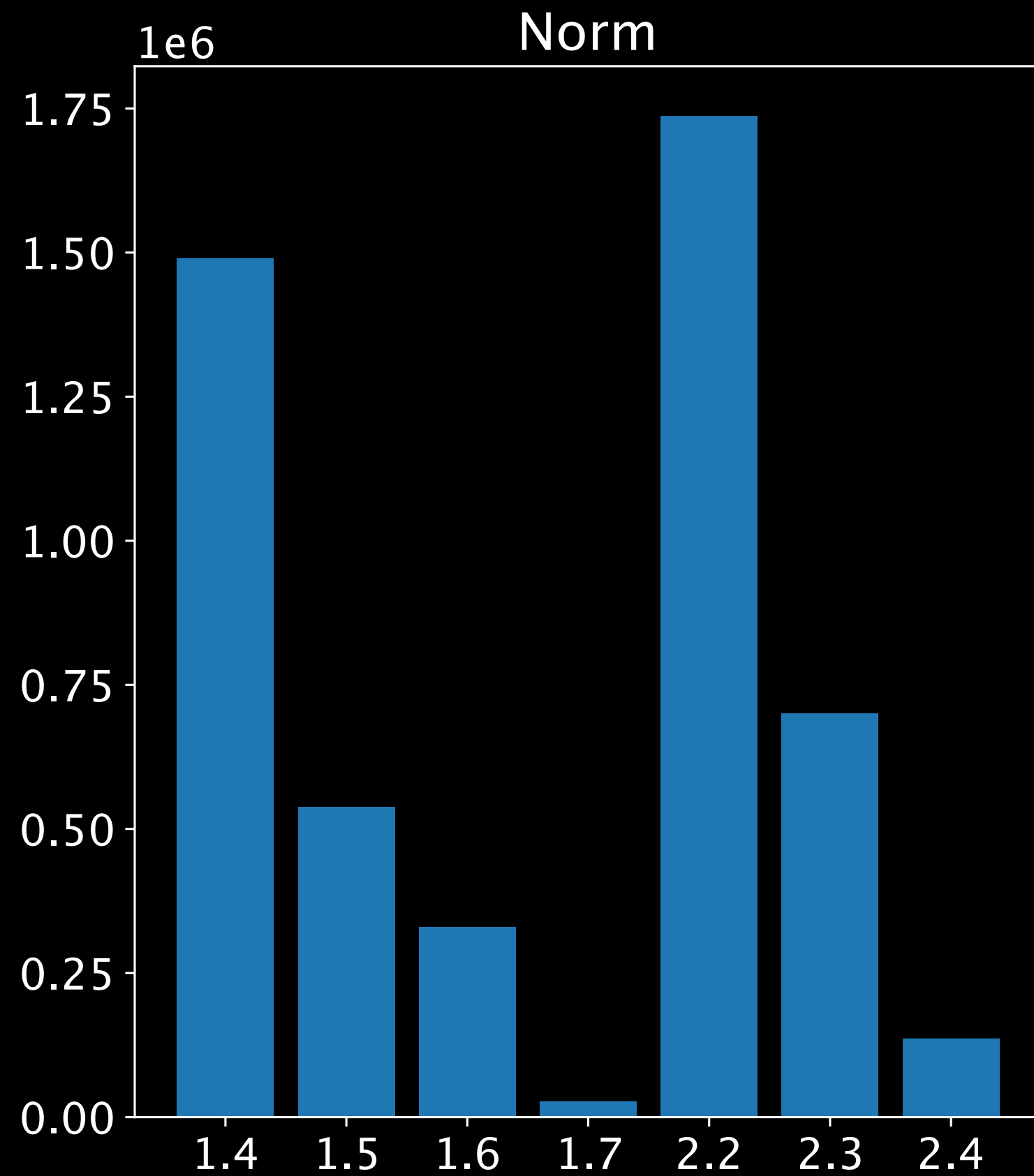
CNN reaches 78% precision (efficiency) and 79% rejection

New Results

Coordinate Distributions

Slow Pions

Distributions of Events for Slow Pions

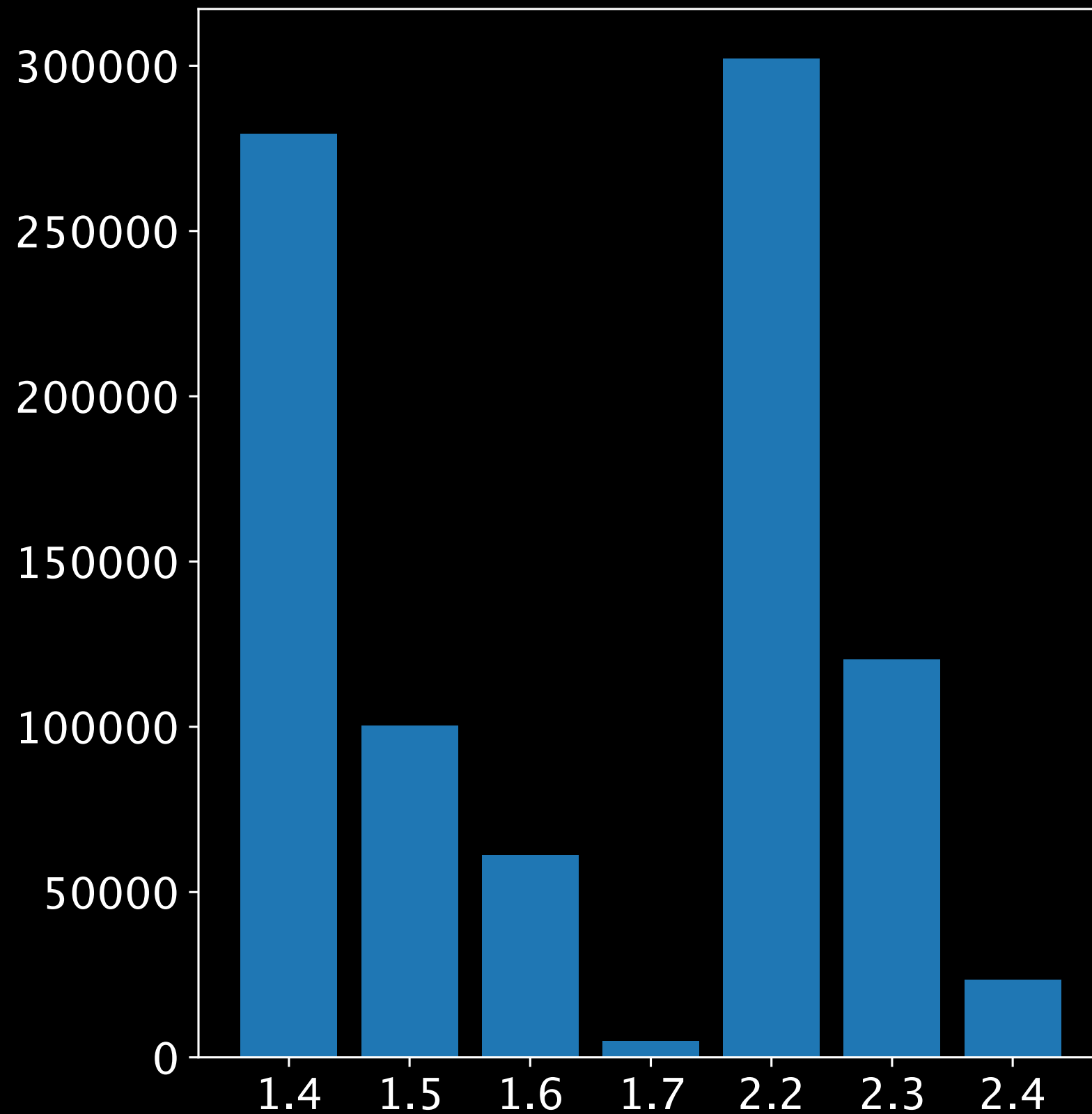


Coordinate Distributions

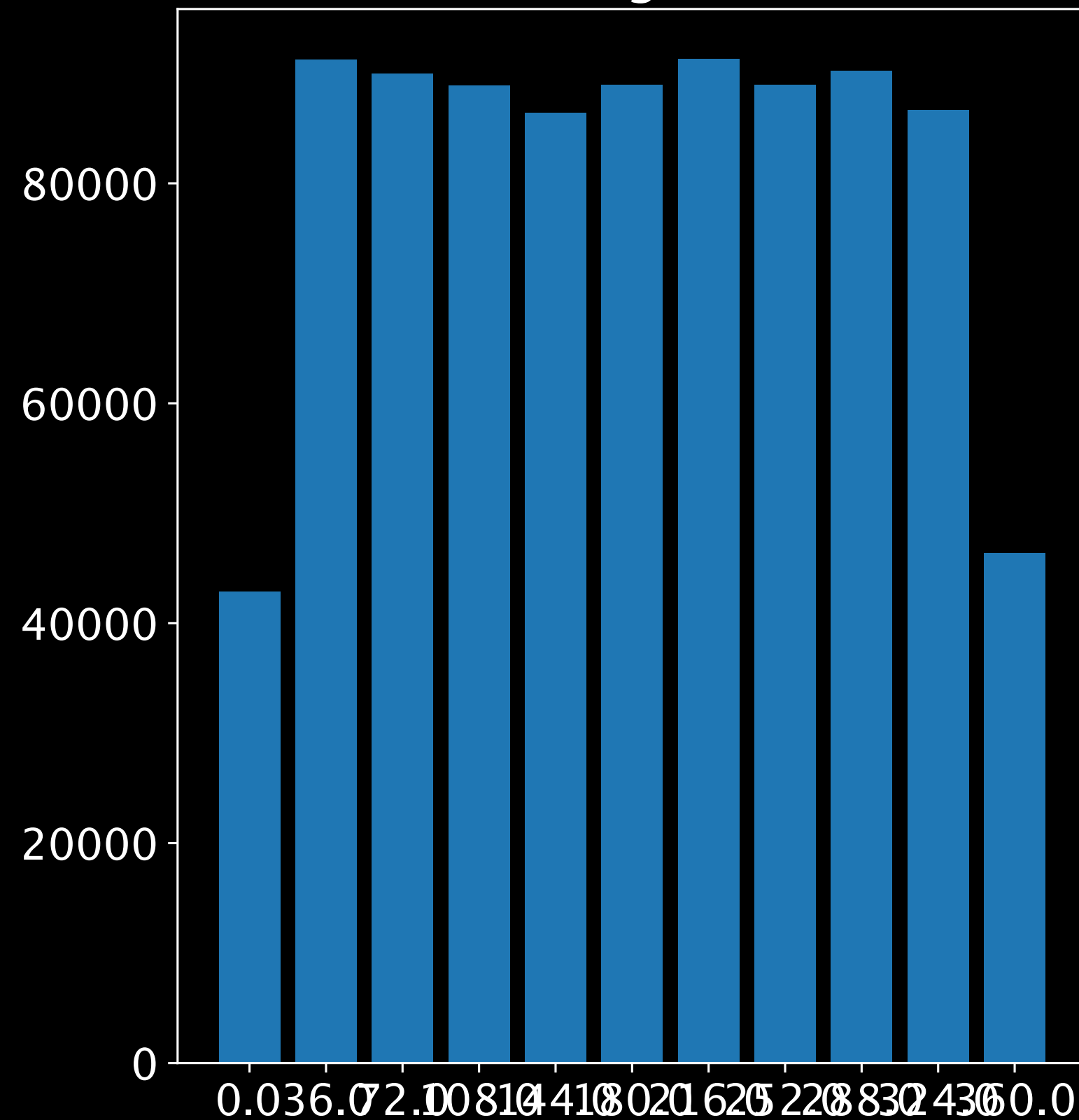
Kaons

Distributions of Events for Kaons

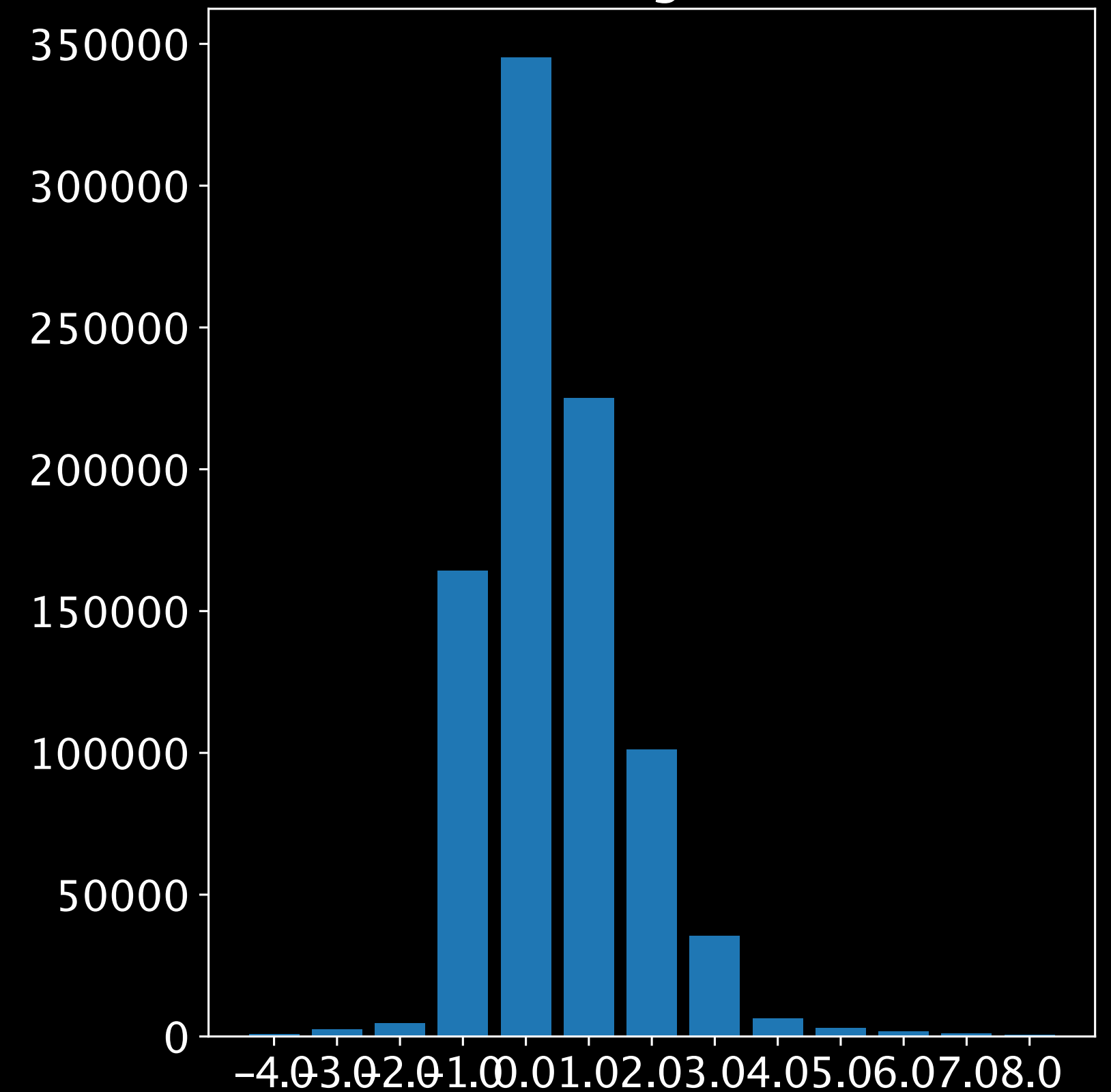
Norm



Angle

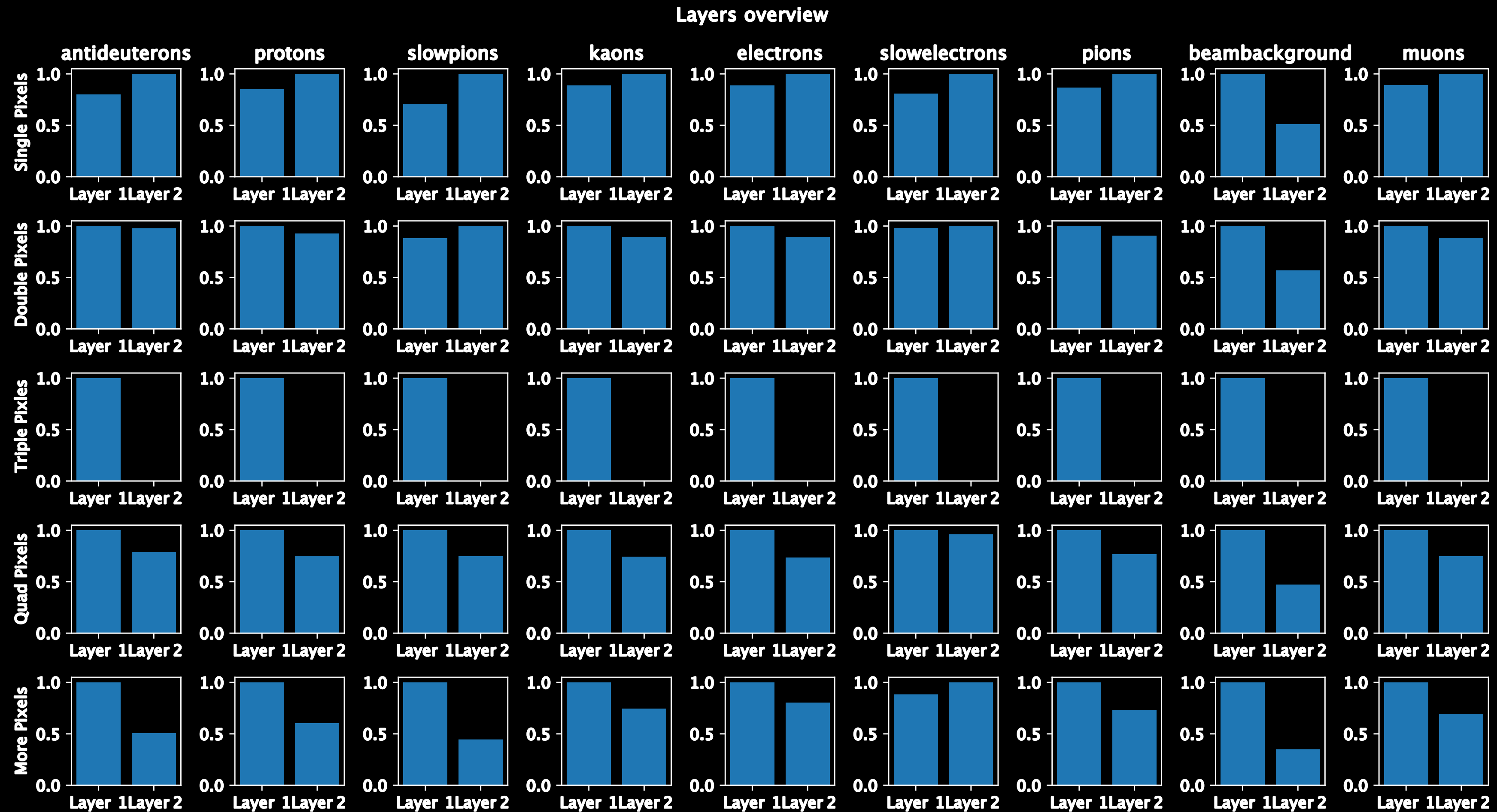


Height



Layer Distribution

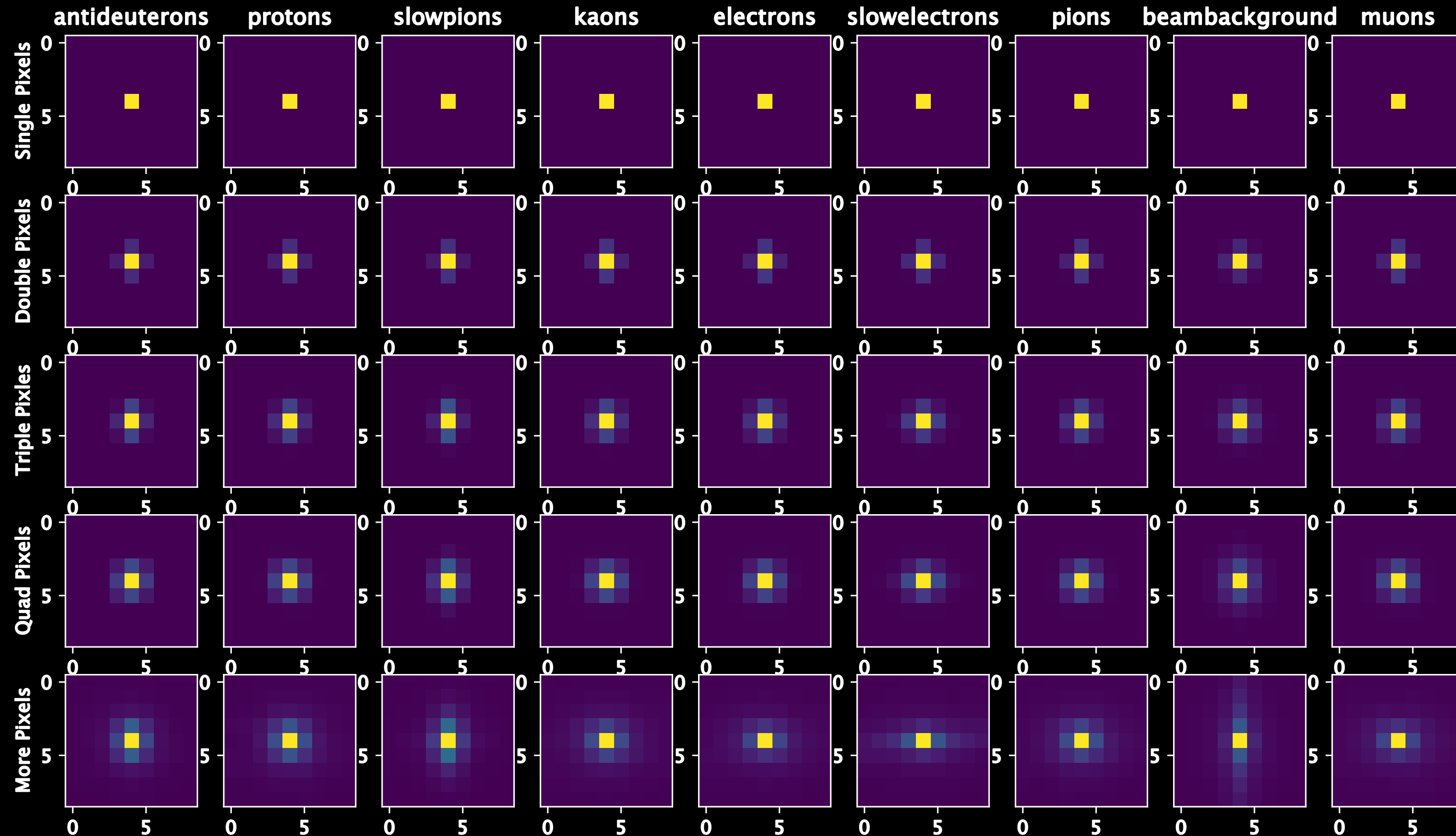
From event coordinates



9x9 pixel matrix

Overview of averaged events

Event overview



Preliminary Results

Preliminary Results

Including Coordinates

- Four test runs
- Excluding coordinates
- Including the coordinates of each Event
 - Cartesian Coordinates
 - Polar Coordinates
 - Just Radius

Preliminary Results

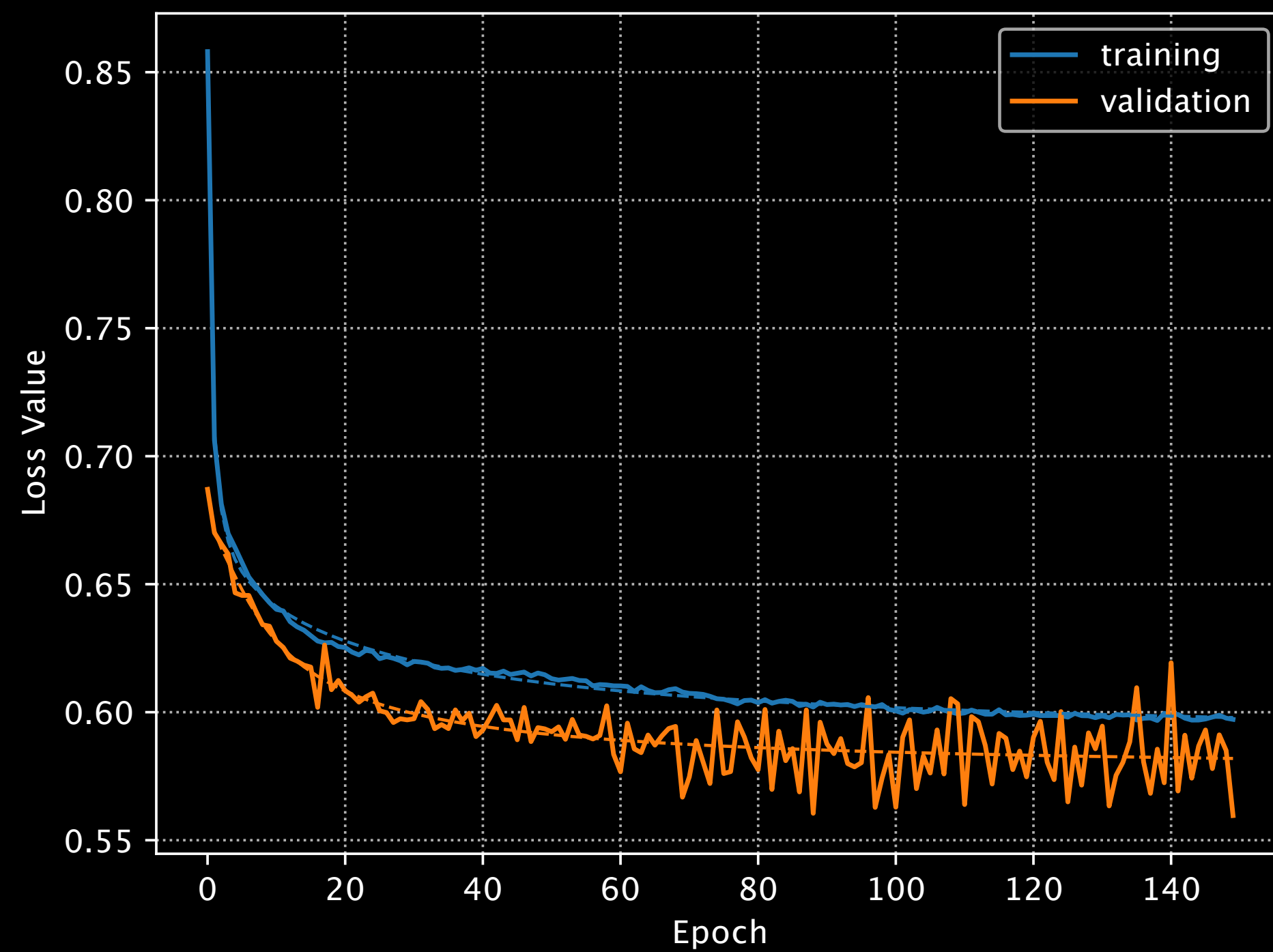
A simple Network

- Simpler and purely linear network
- Seven linear layers
 - 35% dropout rates per Layer
 - ReLU as an activation function
 - Xavier uniform initialised weights
- RMSprop as an optimiser
 - and a reduced learning rate on plateaus
- 150 Epochs with 64 batch size
- Just Slow Pions vs. Kaons
 - 100.000 events from each category

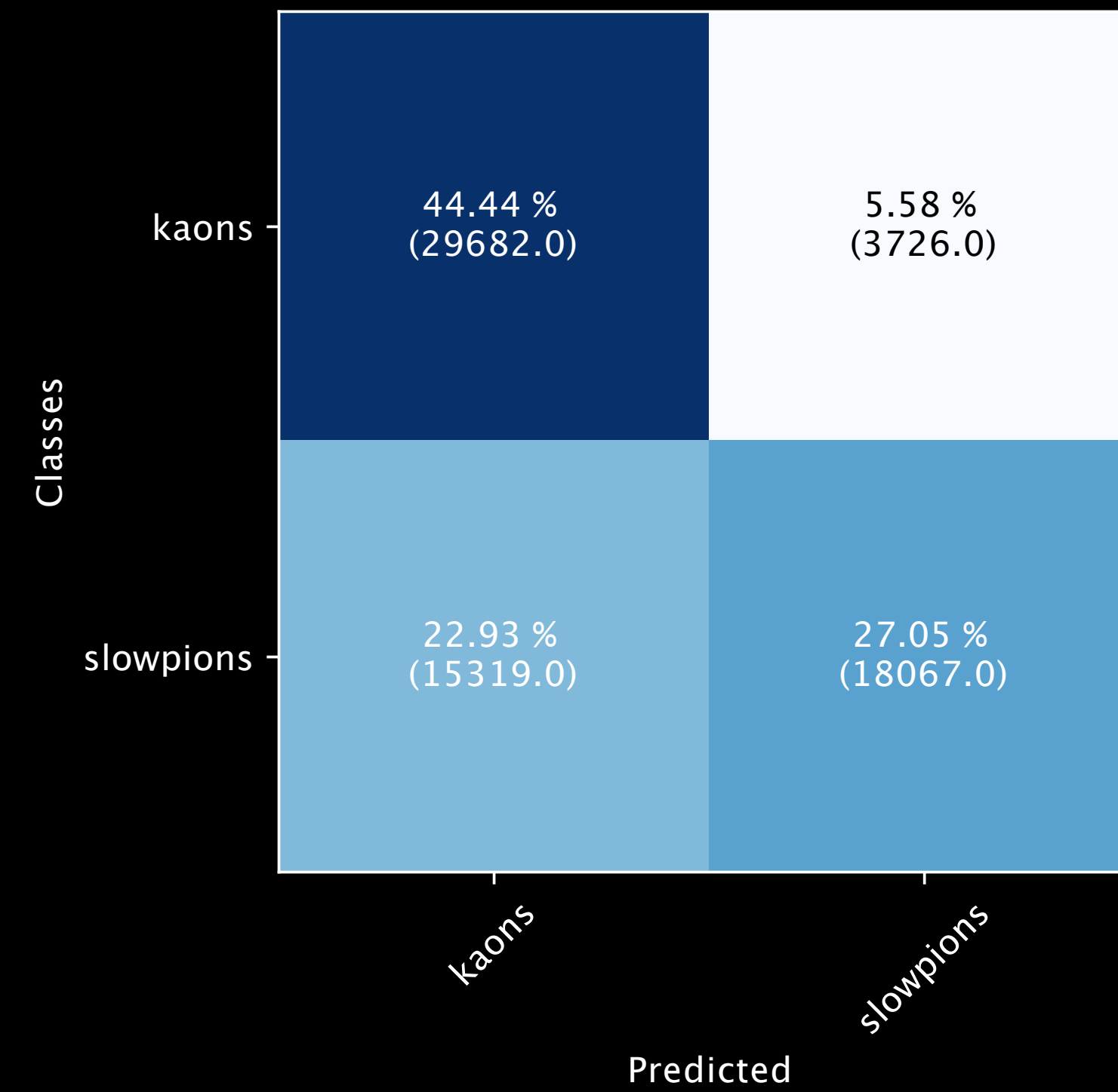
Preliminary Results

Coordinates excluded

Loss Curves



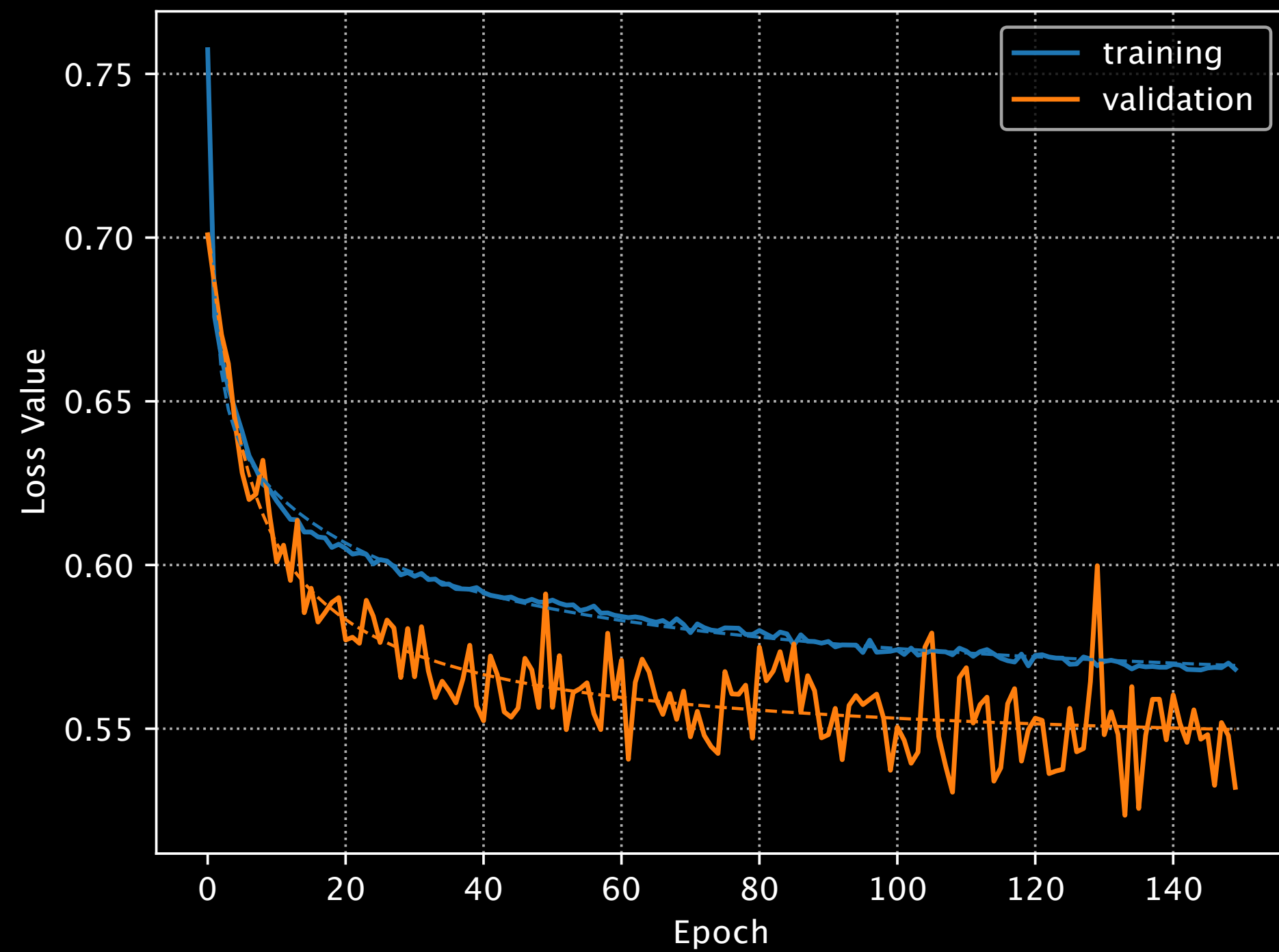
Confusion Matrix



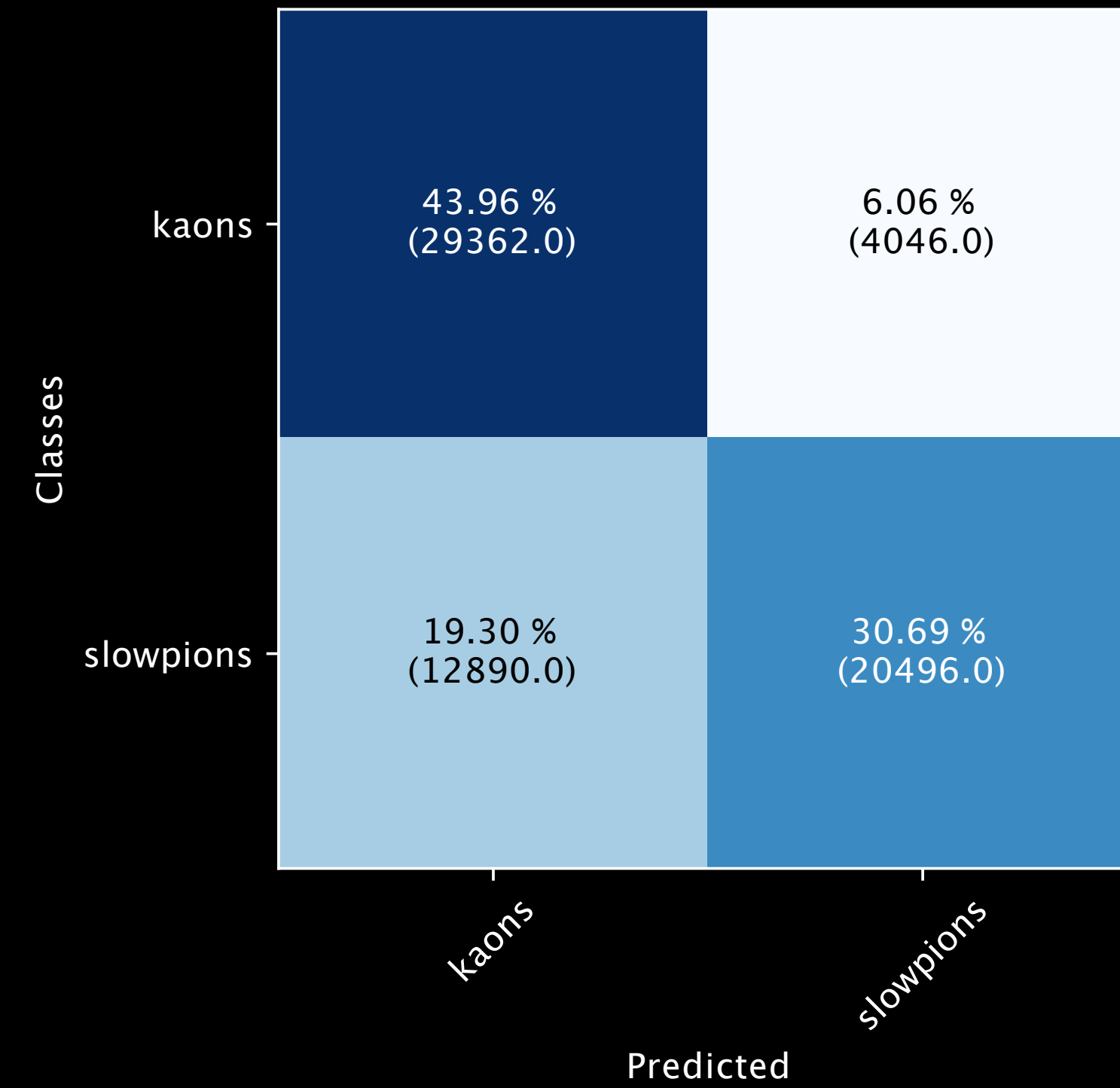
Preliminary Results

Coordinates included

Loss Curves



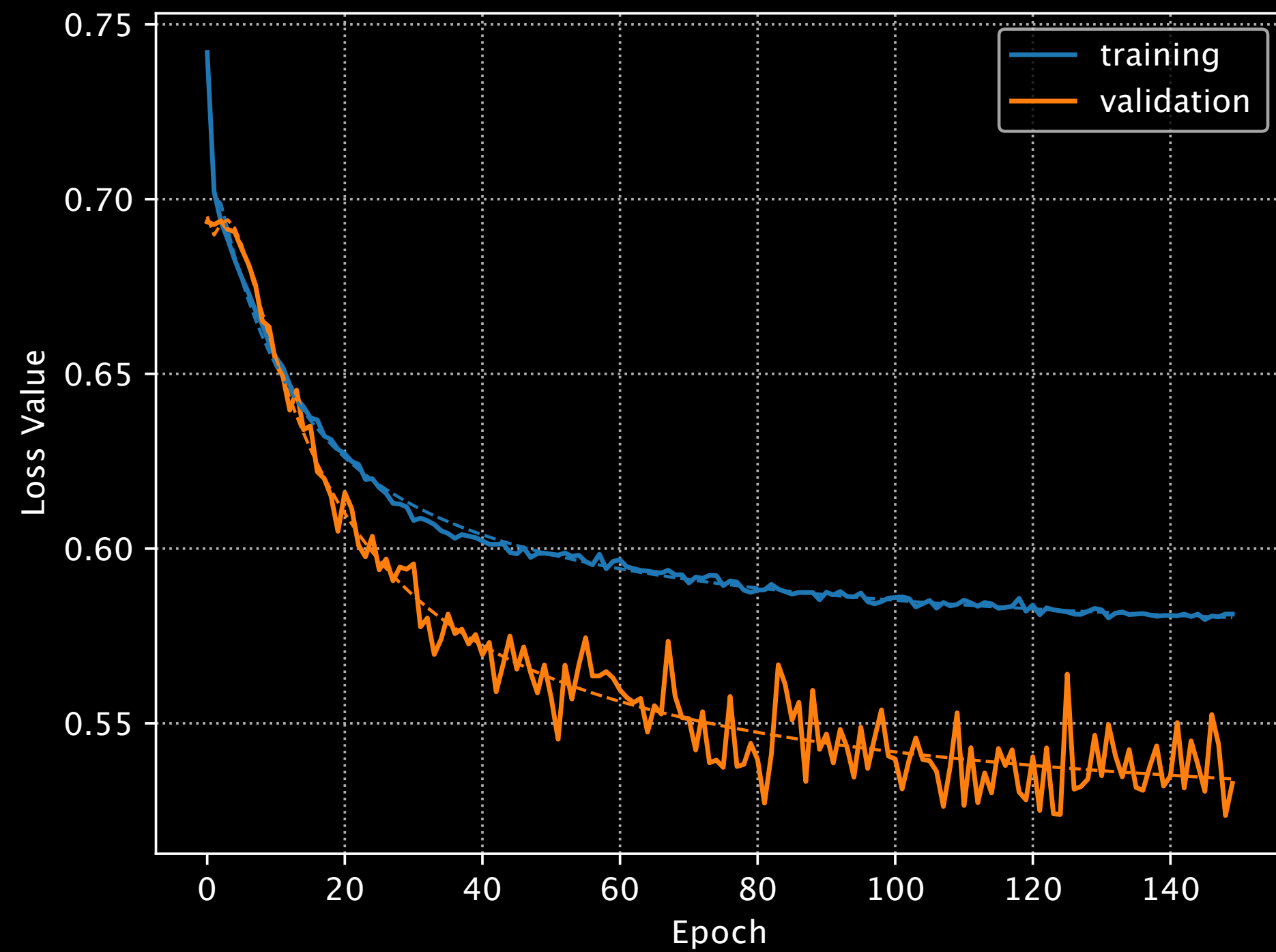
Confusion Matrix



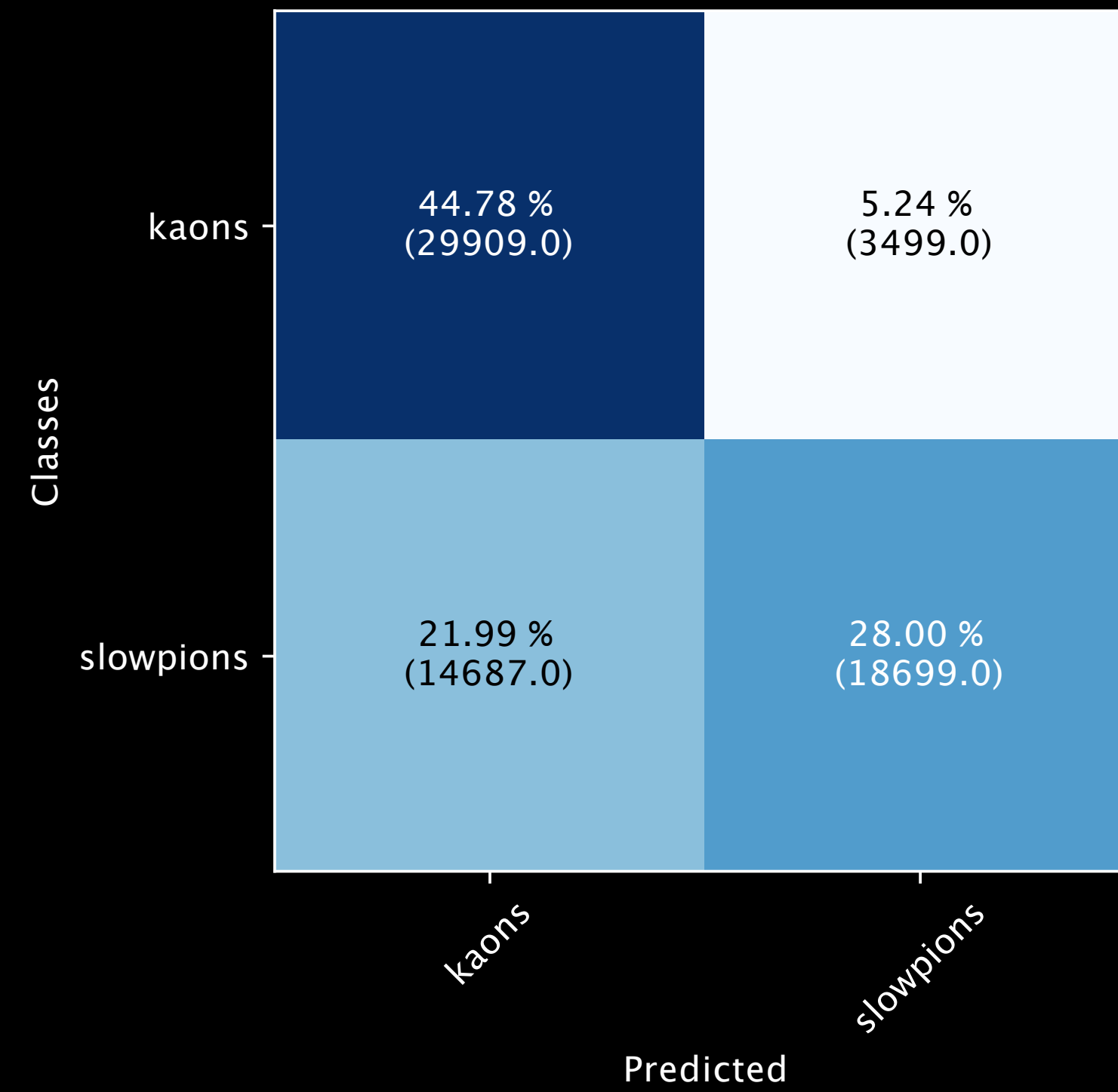
Preliminary Results

Polar Coordinates included

Loss Curves



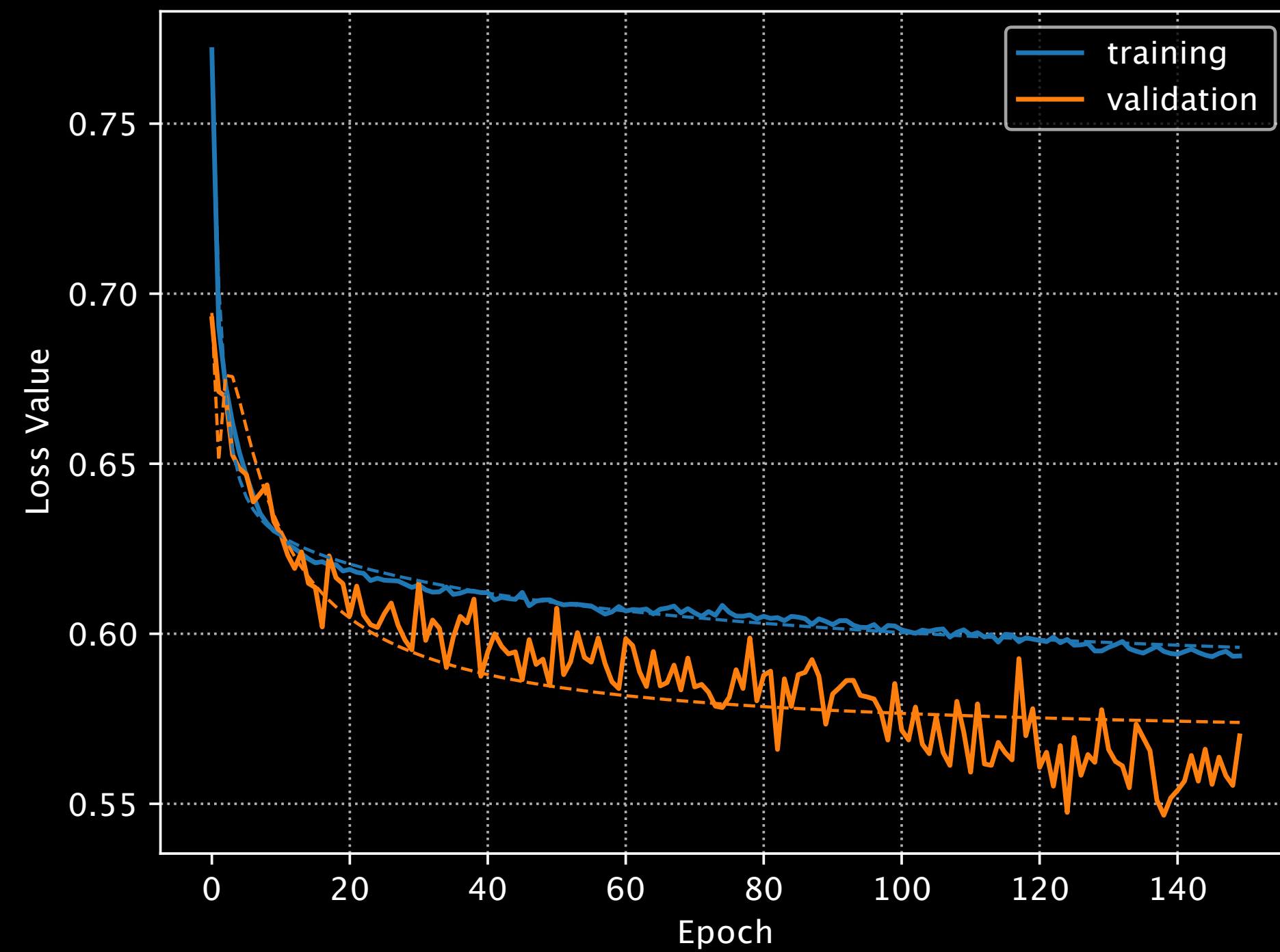
Confusion Matrix



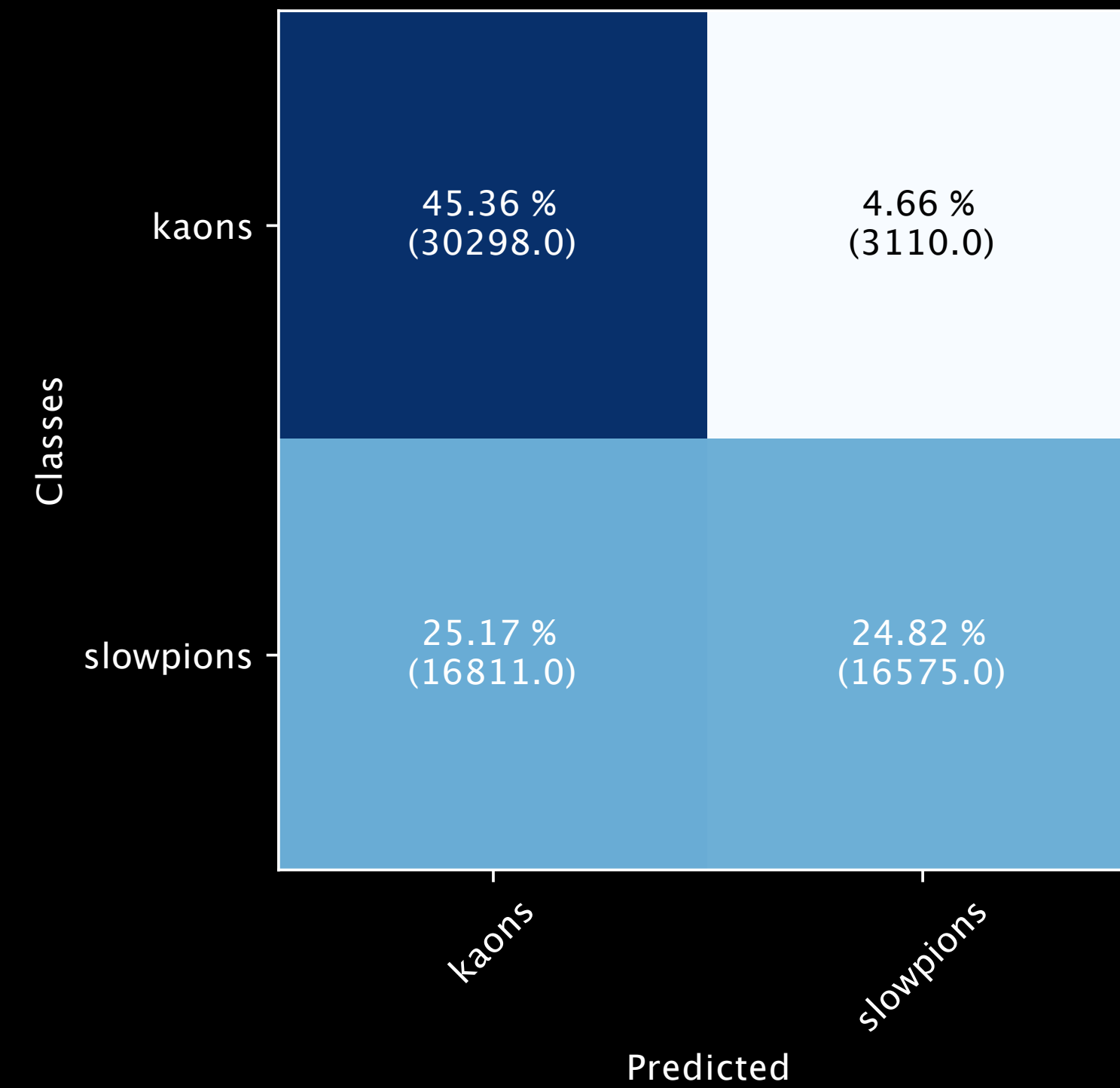
Preliminary Results

Only radius included

Loss Curves

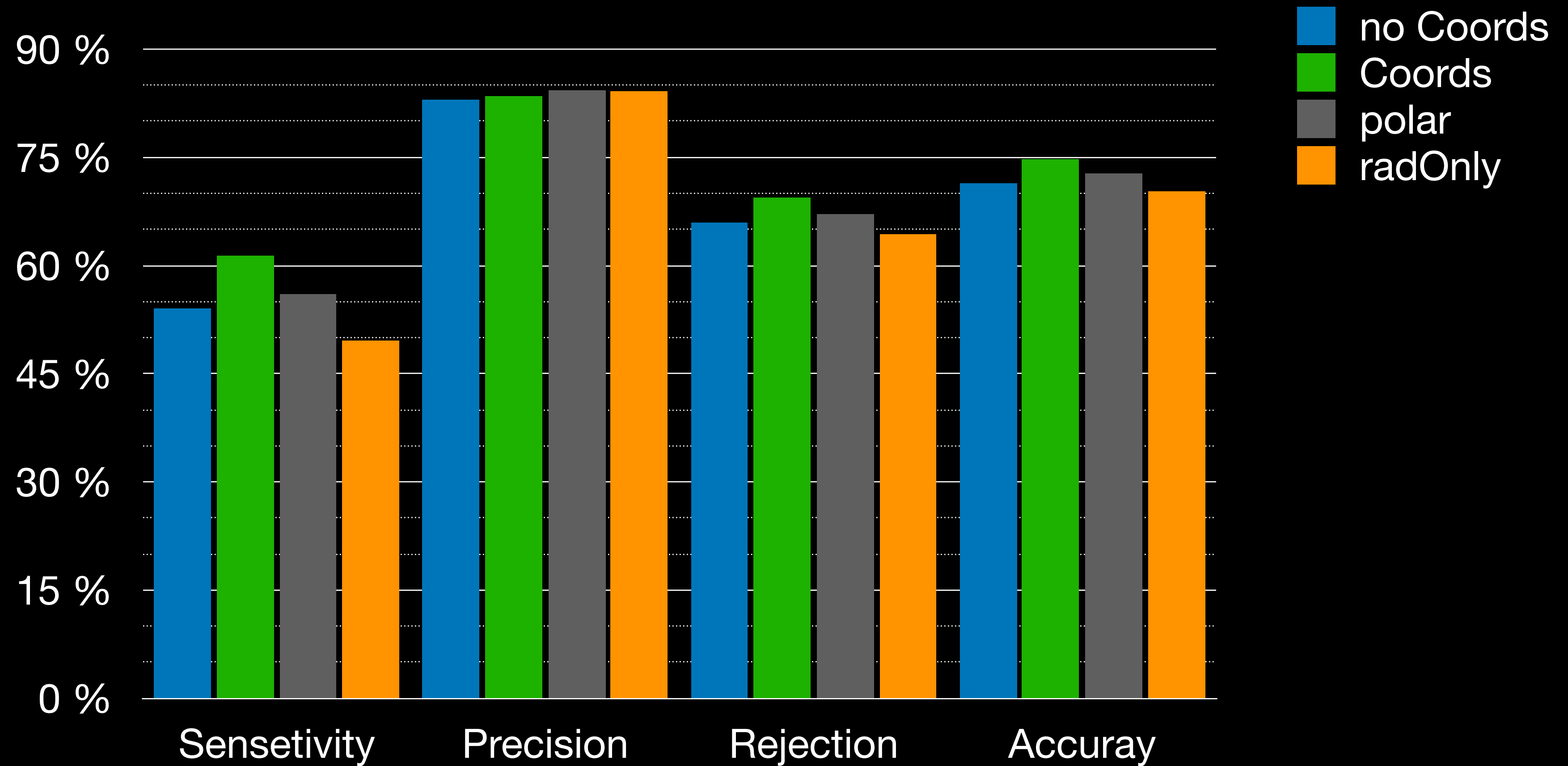


Confusion Matrix



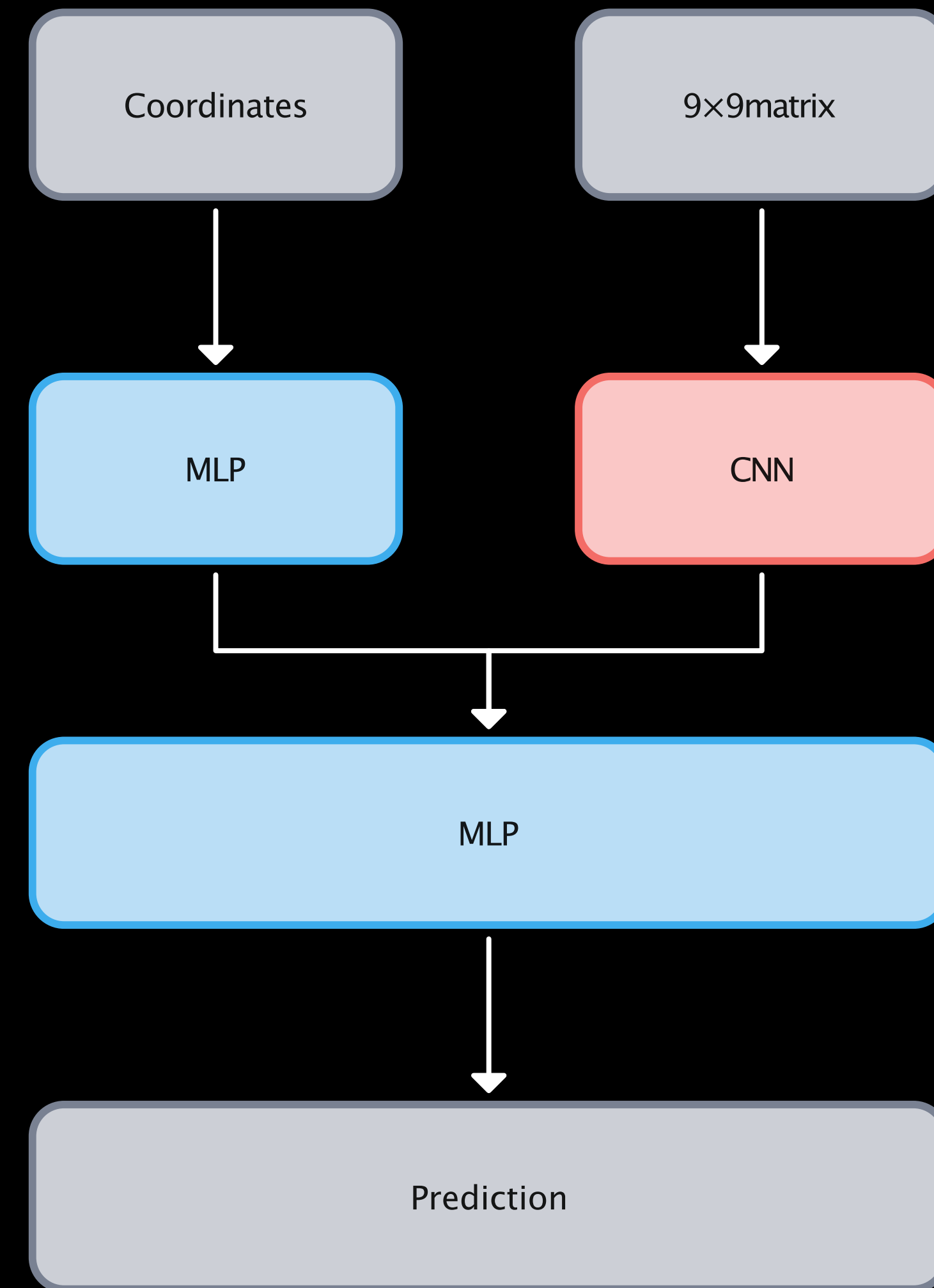
Preliminary Results

Summary



Summary & Outlook

- improved performance with coordinates
- better generalisation as seen in loss curves
- implementing with a CNN
 - CNN for image part
 - MLP for coordinates
 - combined MLP at the end



Thank you for your attention