GNN LST

Input feature plots and T5-NN results

April 21st, 2023

P. Chang, ***J. Guiang***

UC San Diego

UF UNIVERSITY of FLORIDA

Overview

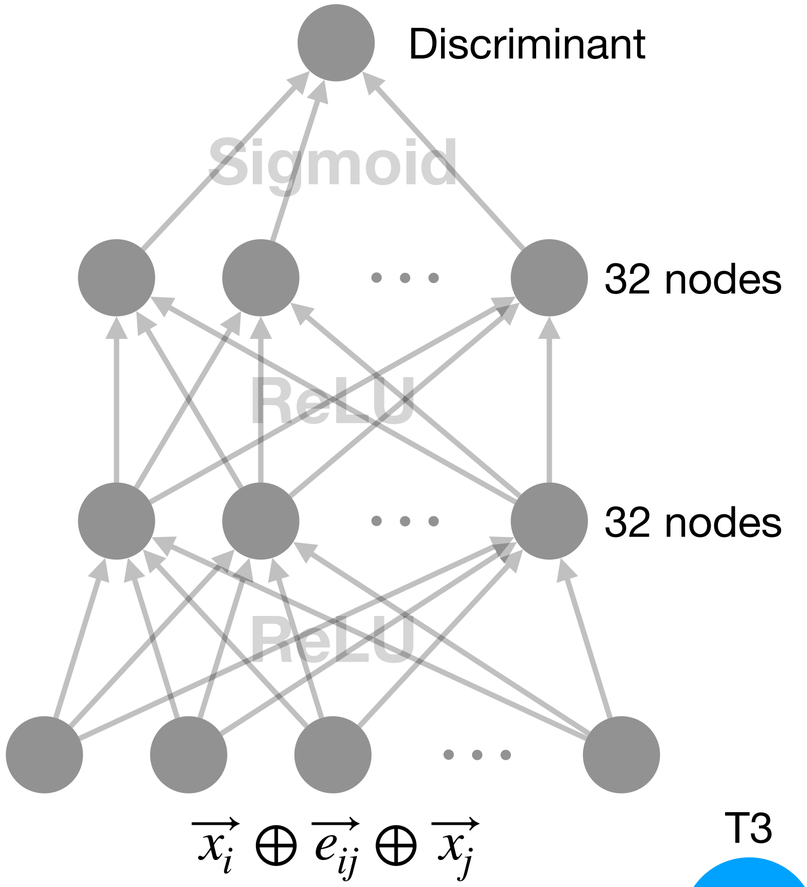
- Training a very simple NN to classify T5s
- First: showing some plots of the input features (right) before and after normalization
  - $p_T \rightarrow \log(p_T)$, radius $\rightarrow$ log(radius), dr $\rightarrow$ log(dr)
- Showing two trials today:
  - Trial 1: classifying T5s *after* χ^2 cuts
  - Trial 2: classifying T5s *without* χ^2 cuts
    - χ^2 not used as an input feature here

Object	Feature
T3	p_T
	Inner anchor hit r, z, ϕ, η, δr, layer
	Middle anchor hit r, z, ϕ, η, δr, layer
	Outer anchor hit r, z, ϕ, η, δr, layer
T5 candidate	p_T, η, ϕ
	χ^2
	Inner T3 radius
	Bridge T3 radius
	Outer T3 radius

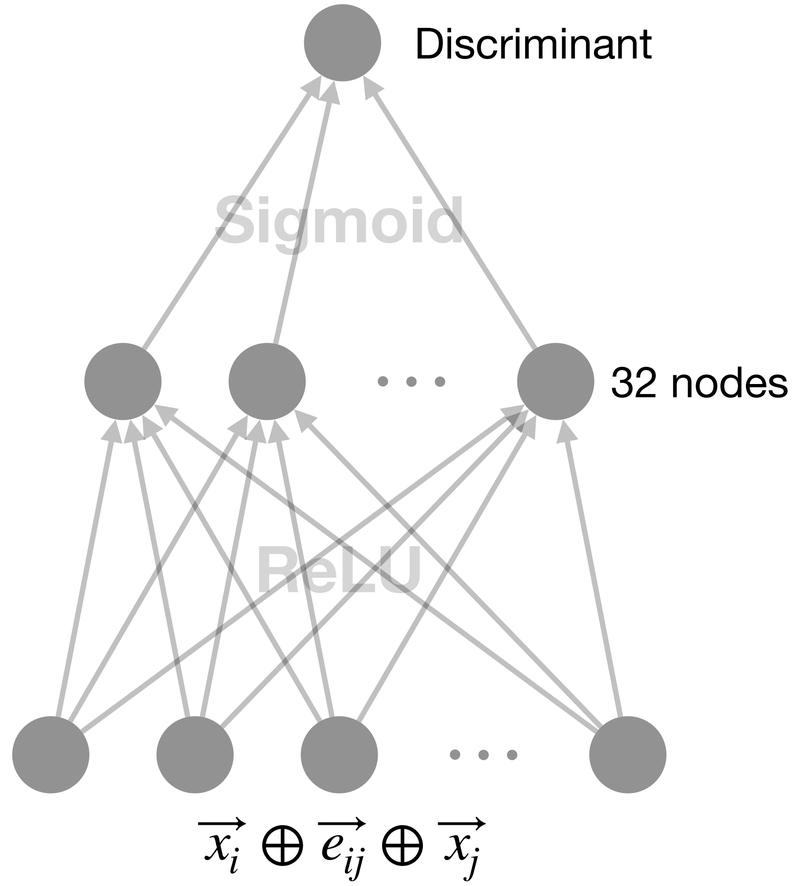
Scaled such that all features $\in$ [0, 1]

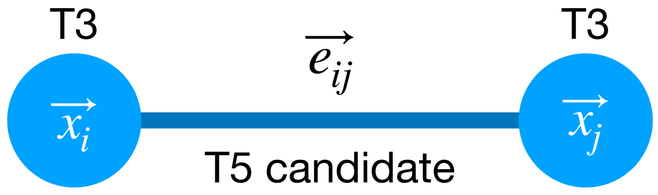
T5 Neural Network

1 hidden layers



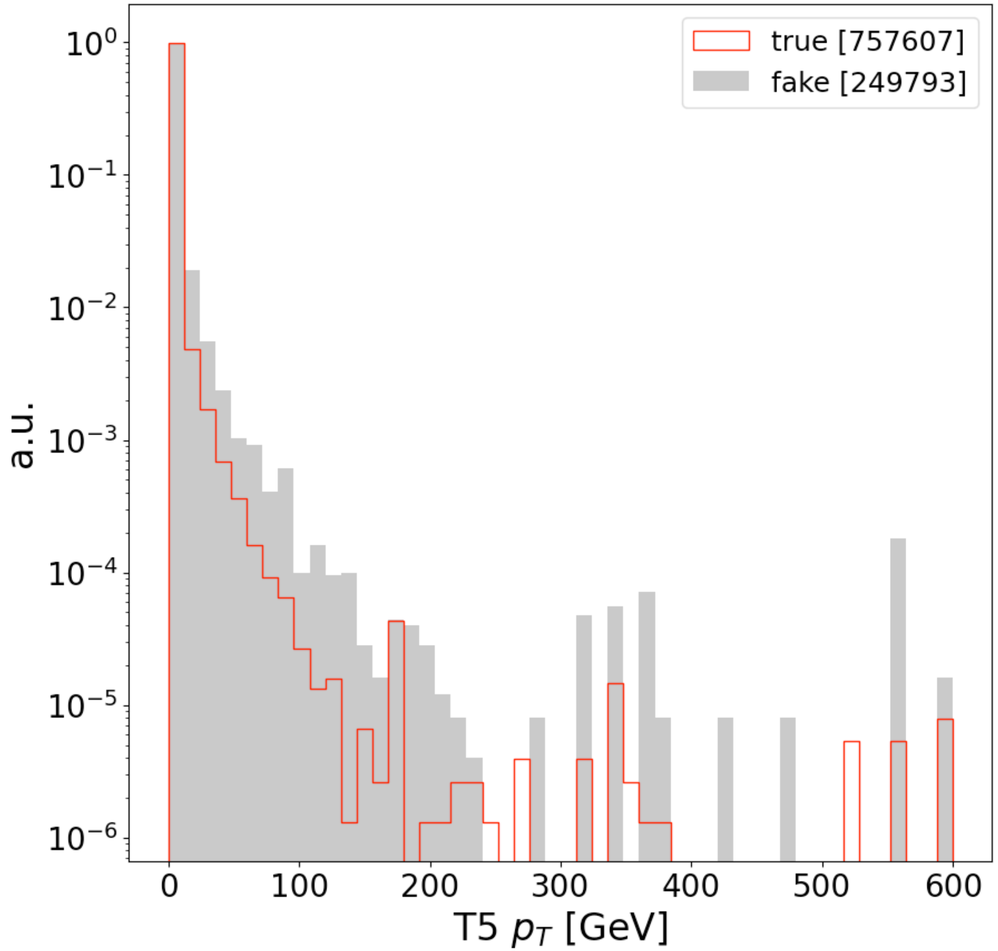
No hidden layers



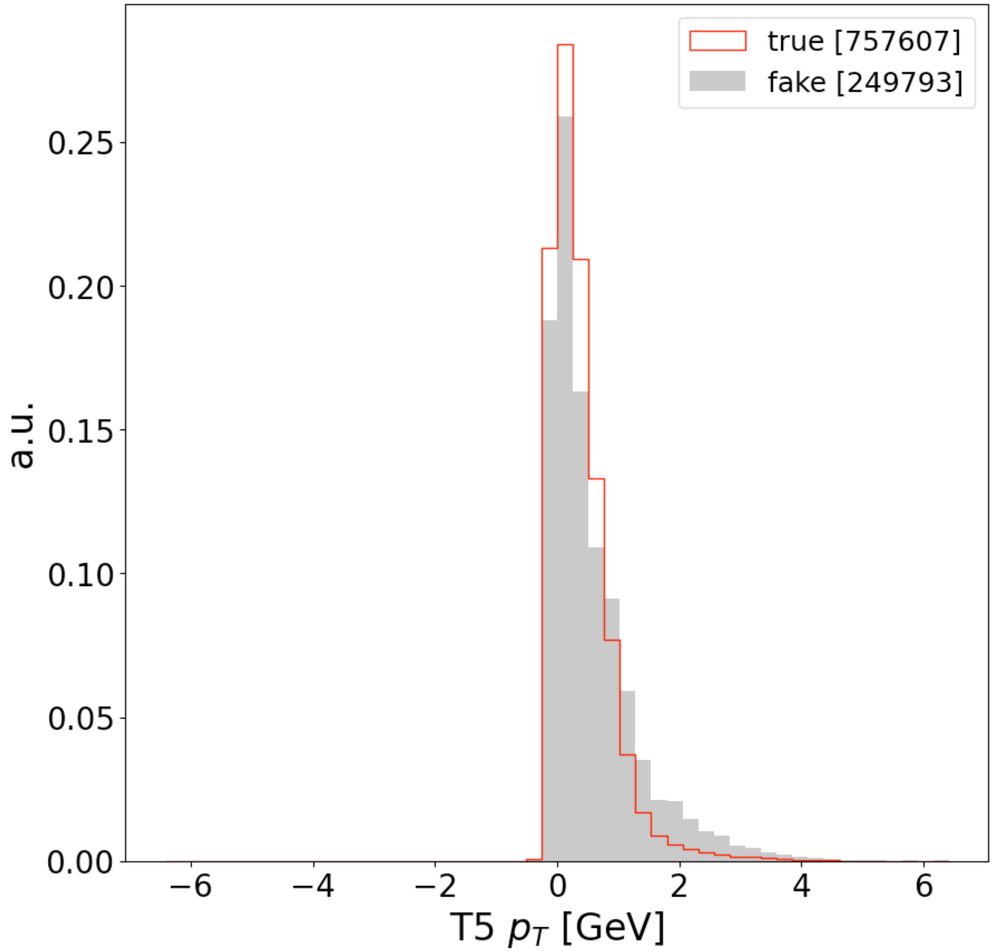


Trial 1: Input Features

All plots available here

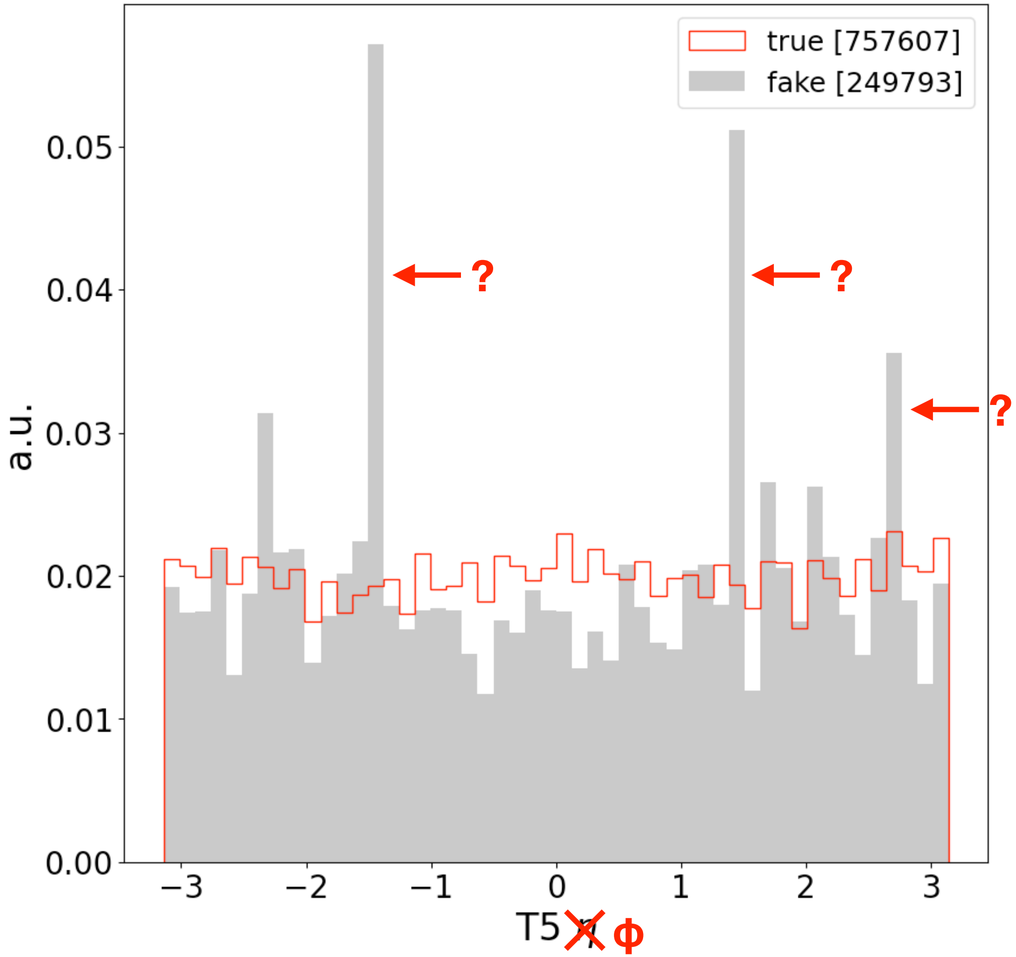


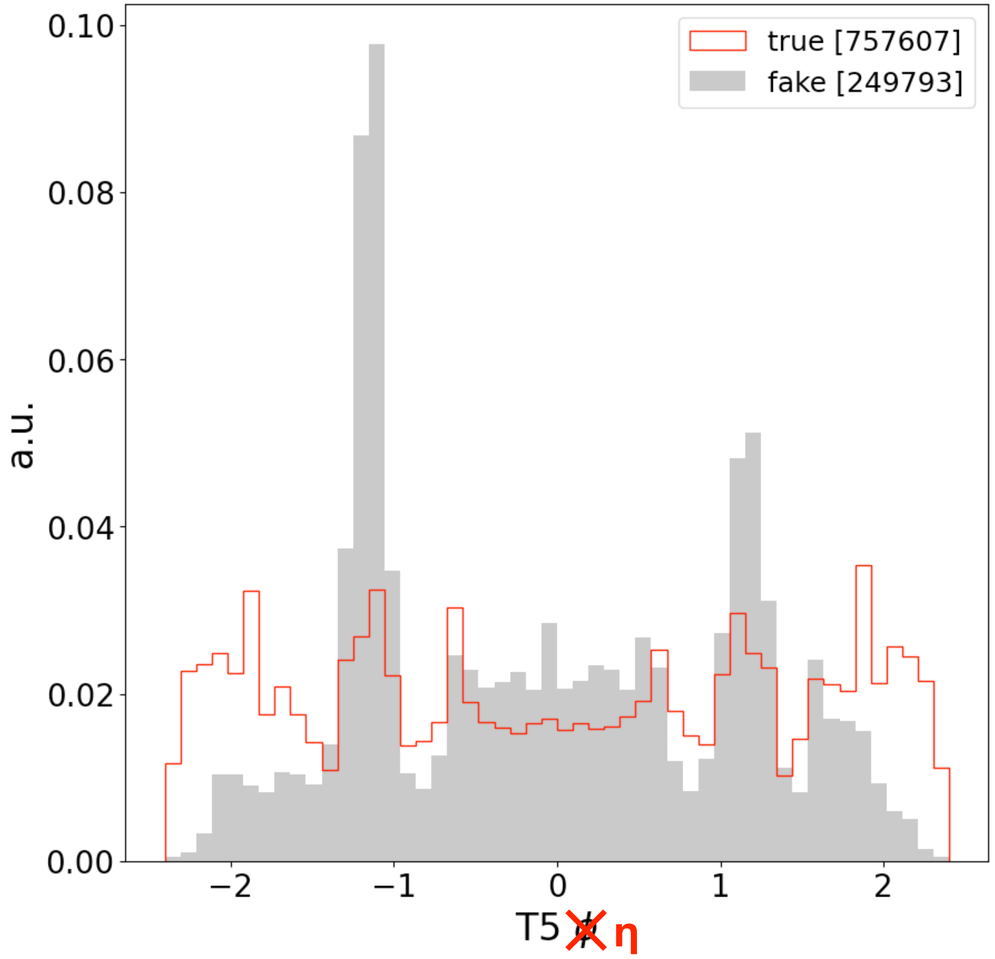
log(x)



Trial 1: Input Features

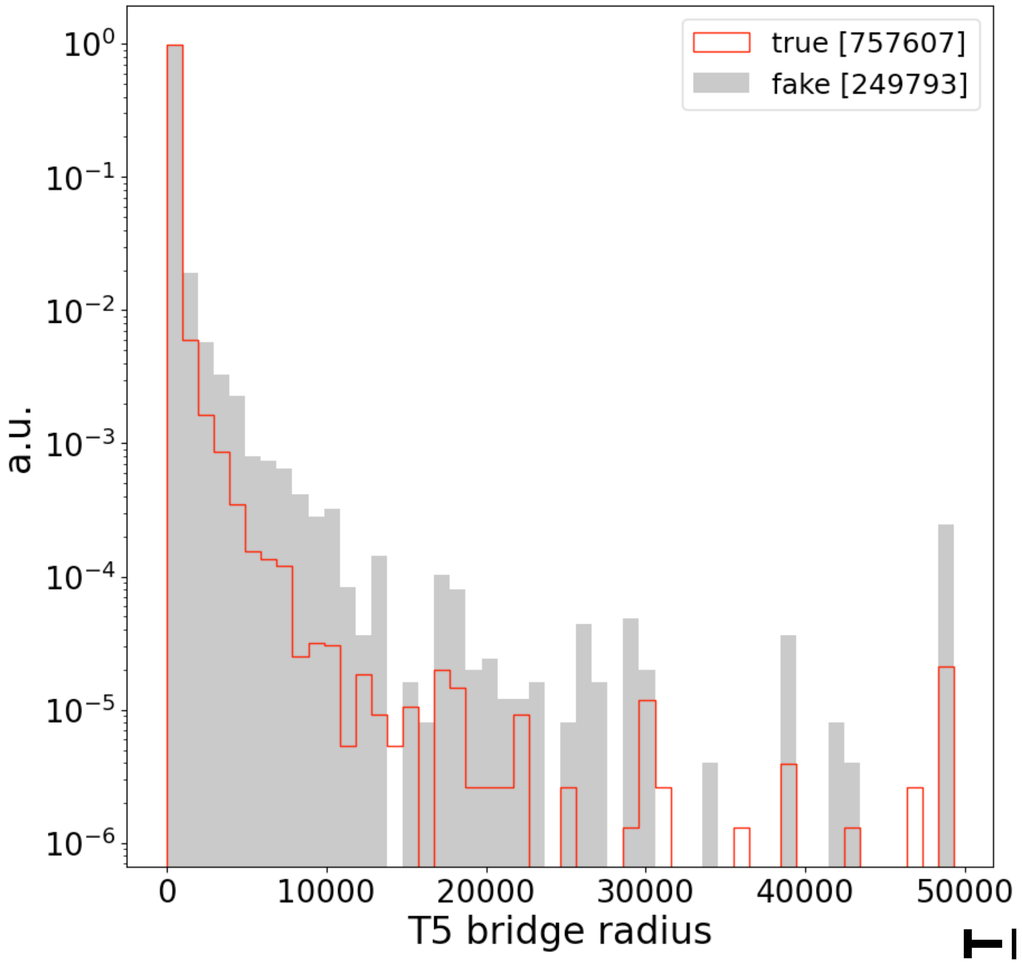
All plots available here



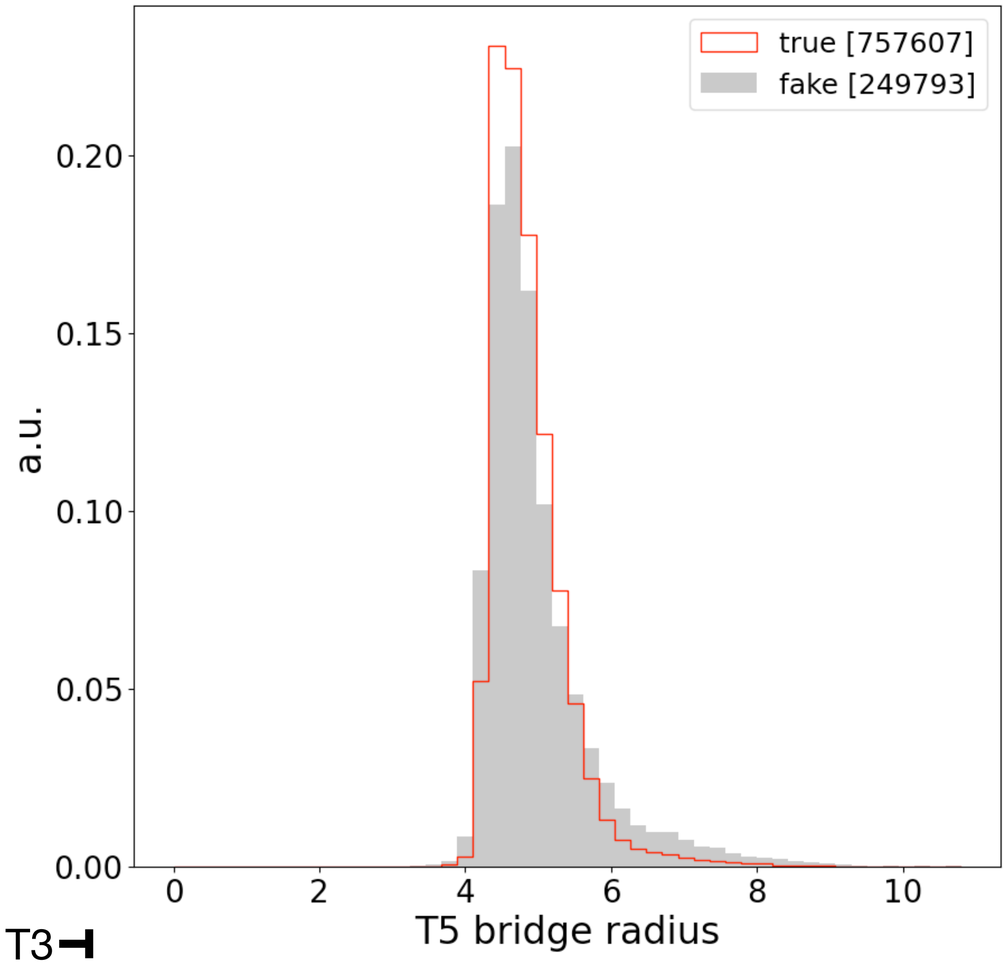


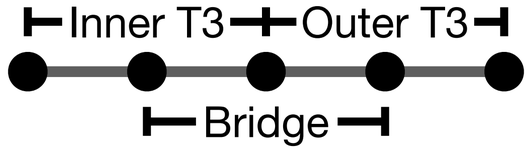
Trial 1: Input Features

All plots available here



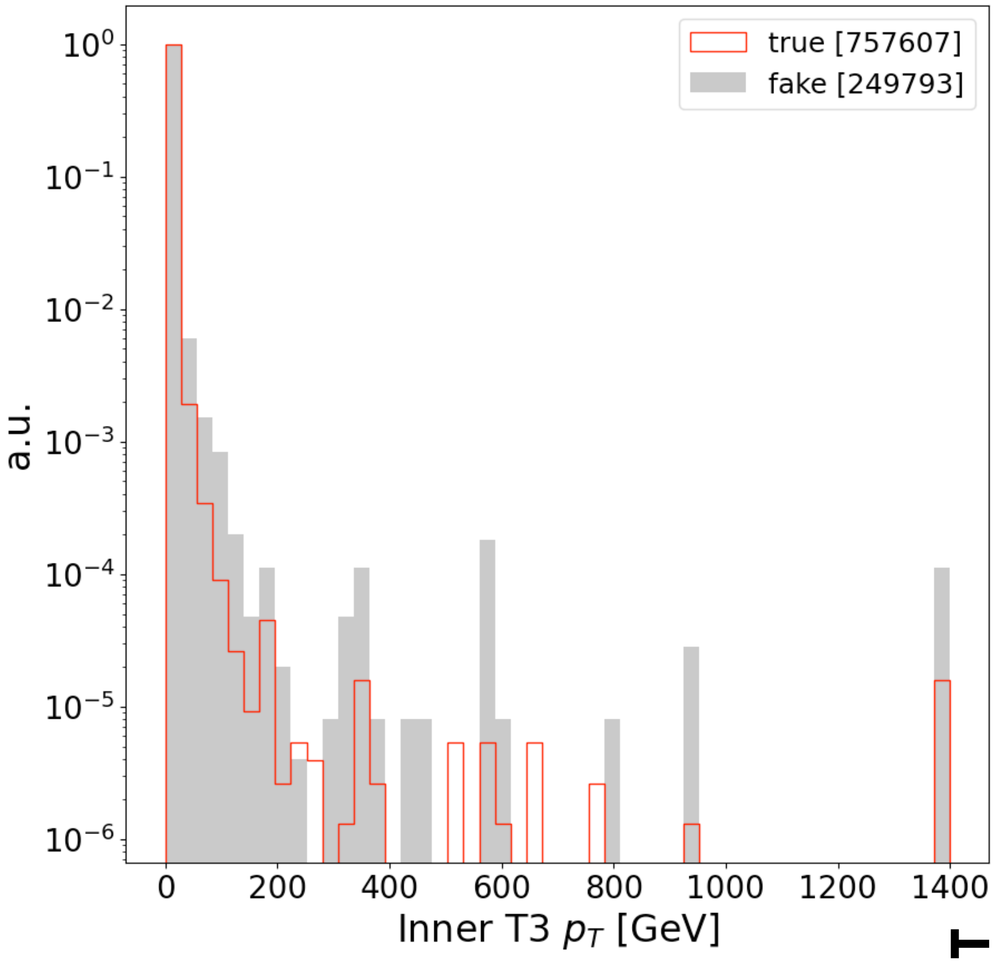
log(x)



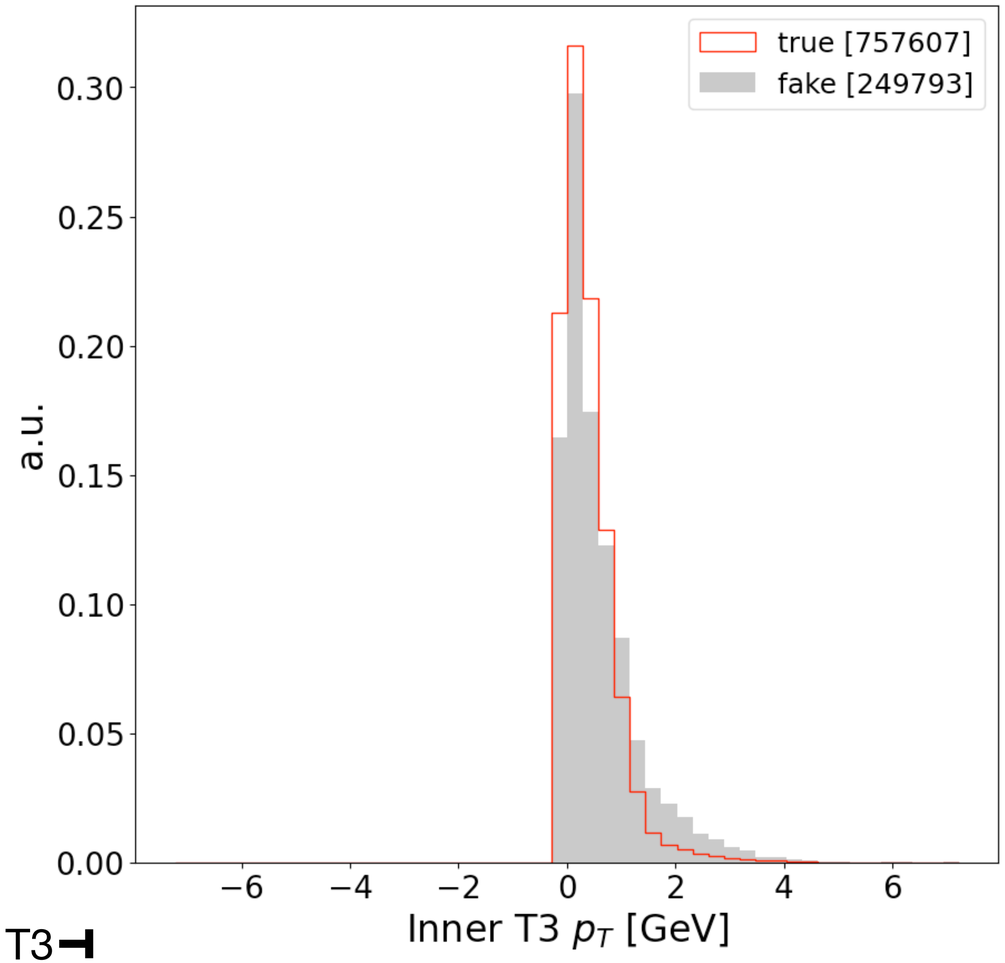


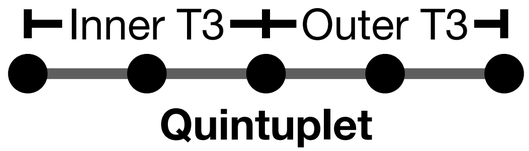
Trial 1: Input Features

All plots available here



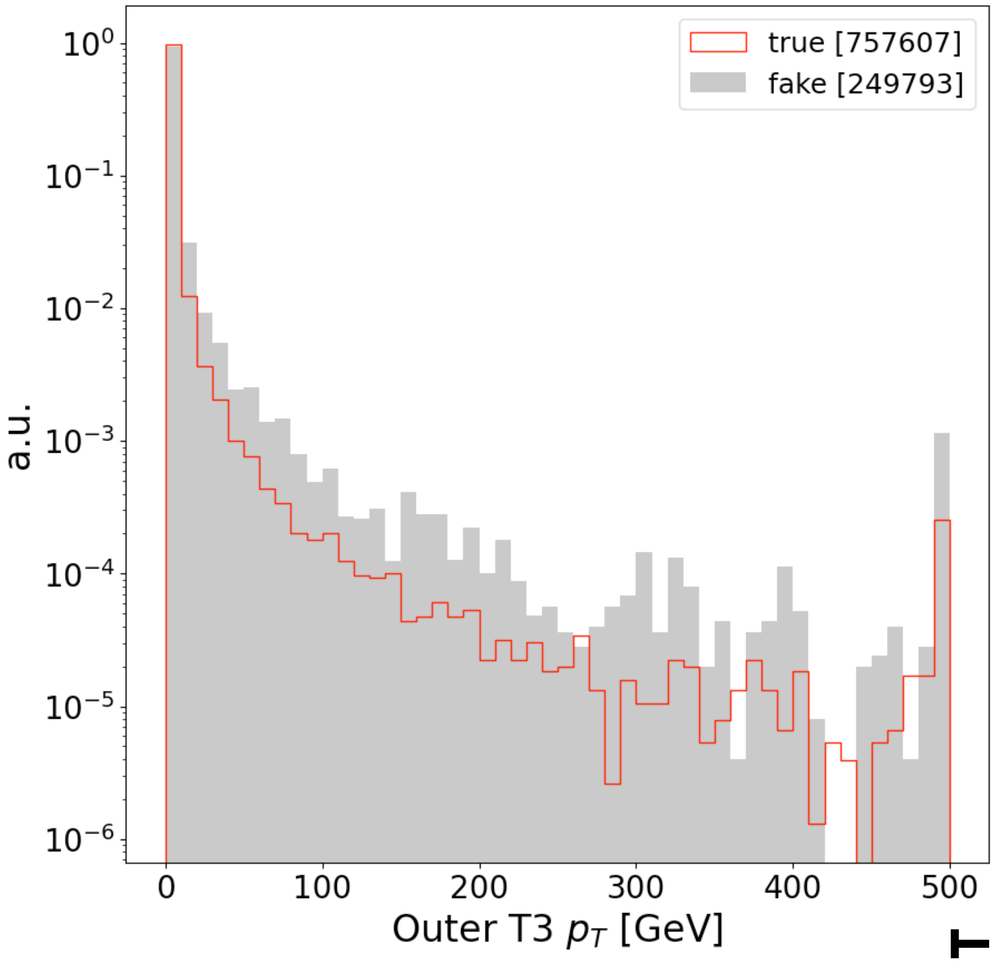
log(x)



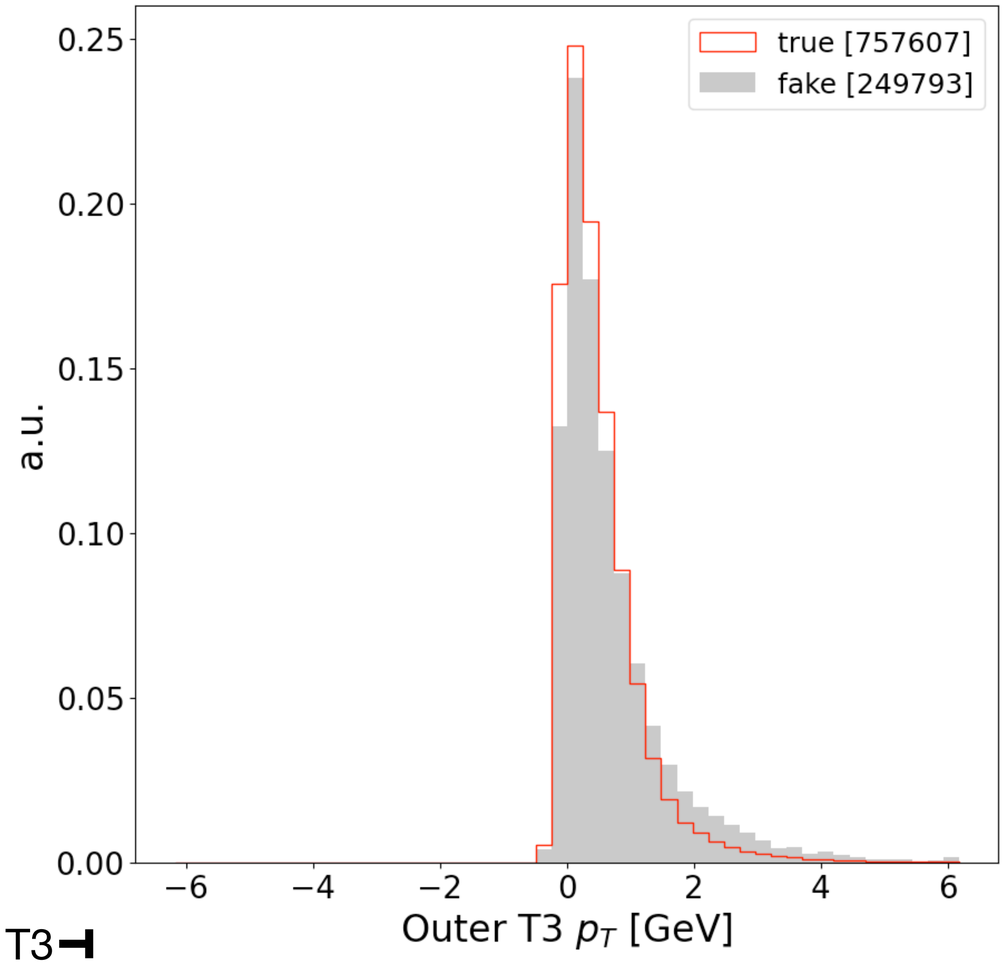


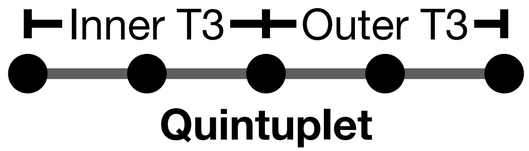
Trial 1: Input Features

All plots available here



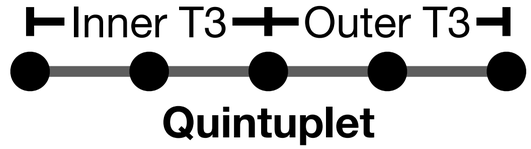
log(x)

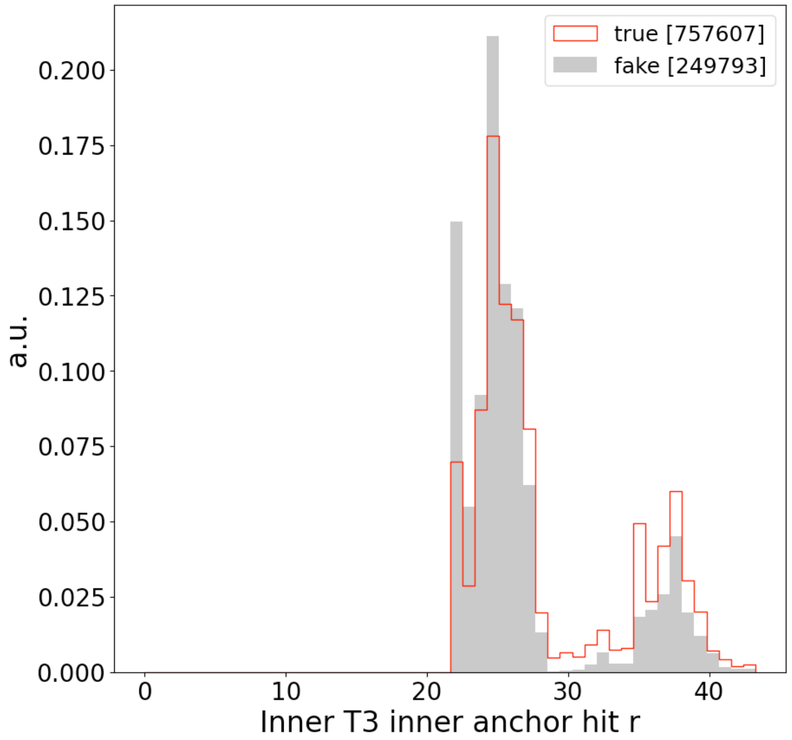


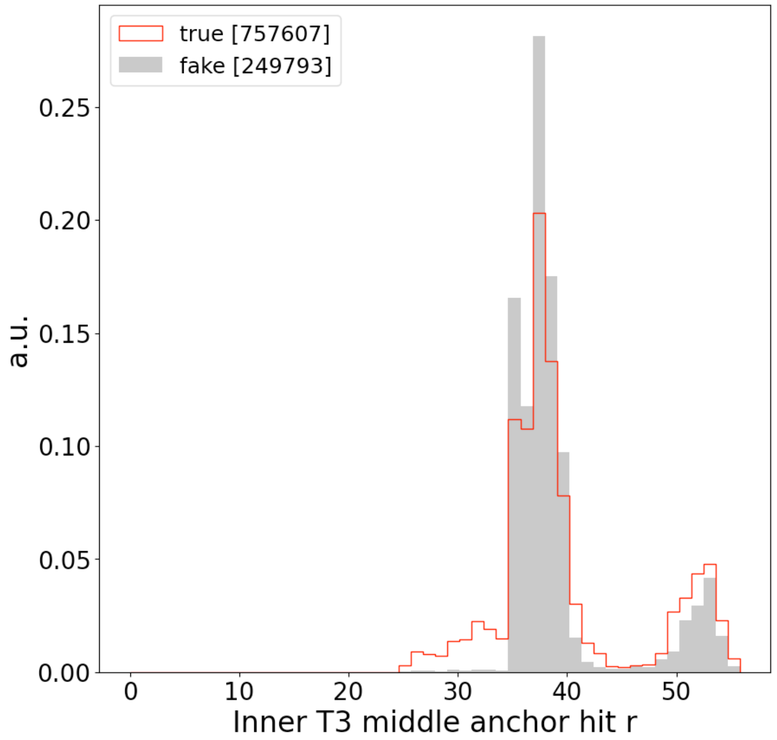


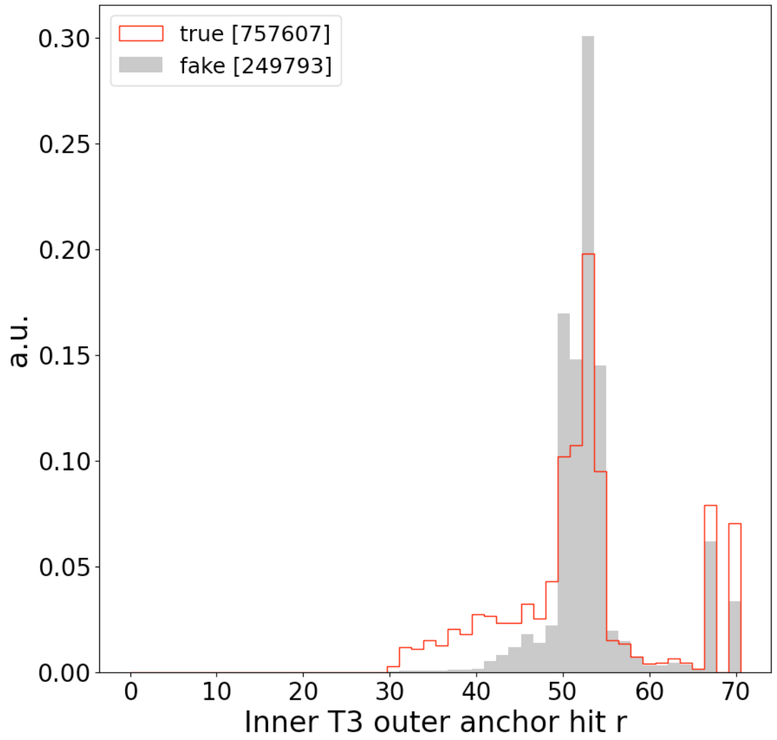
Trial 1: Input Features

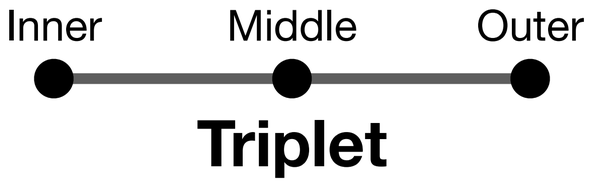
All plots available here





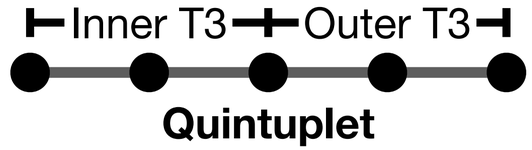


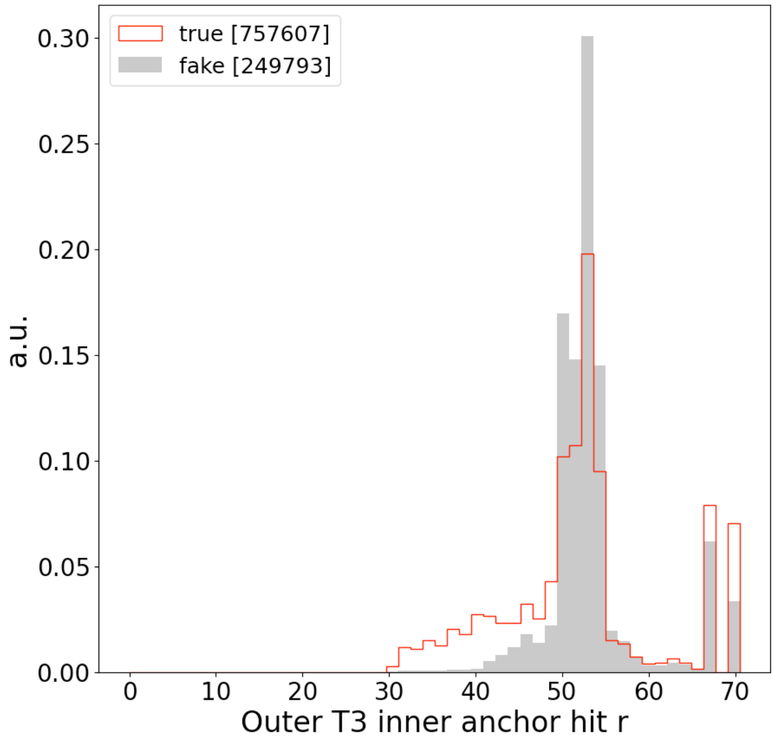


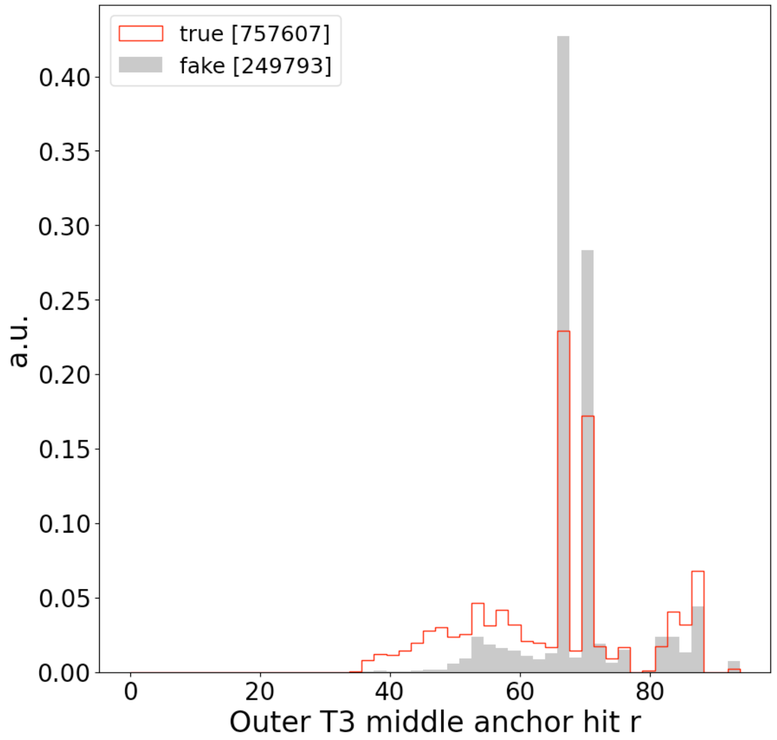


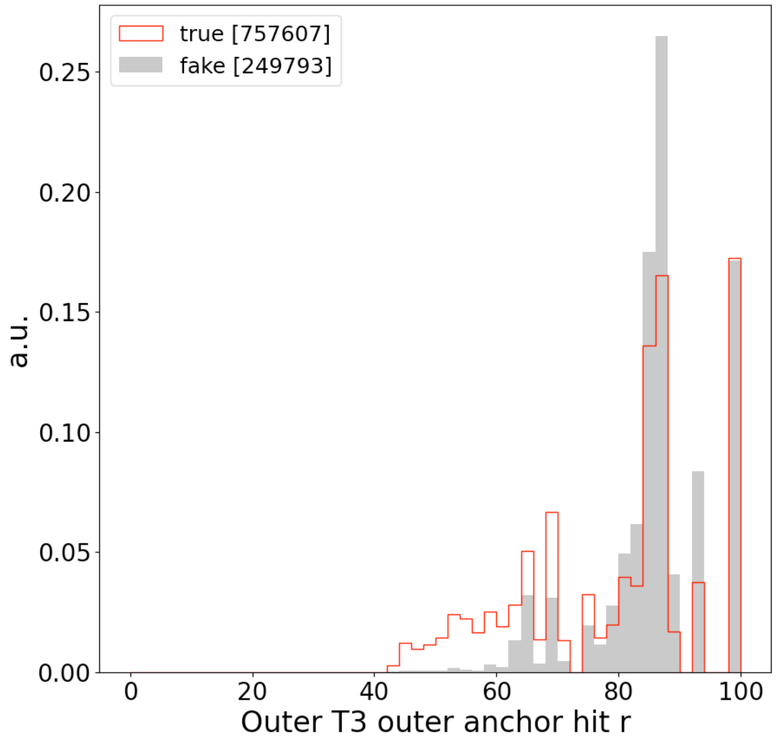
Trial 1: Input Features

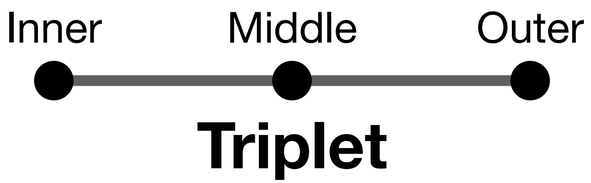
All plots available here





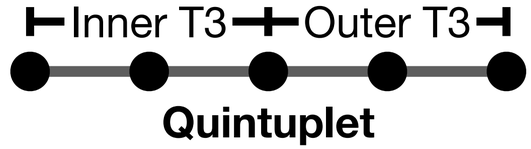


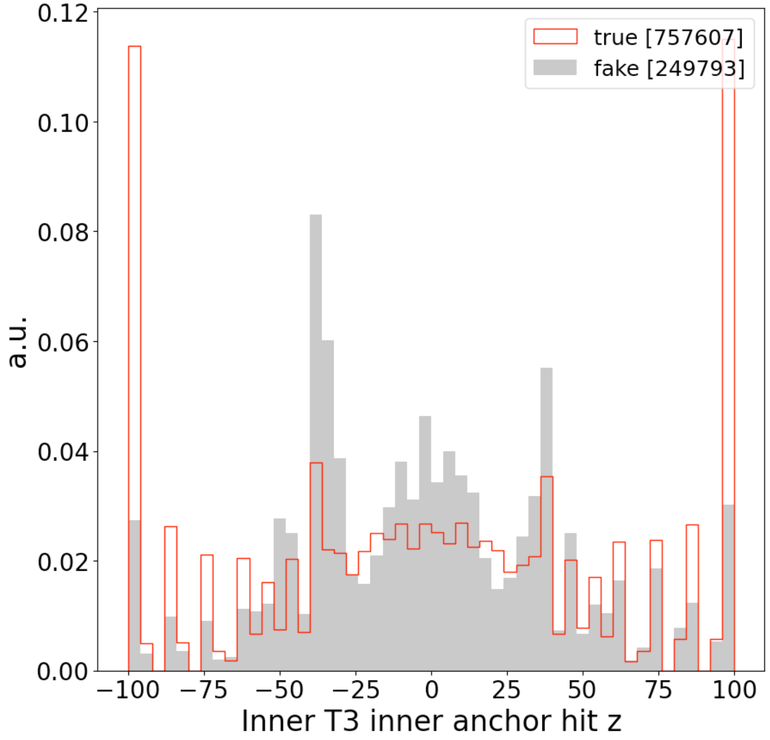


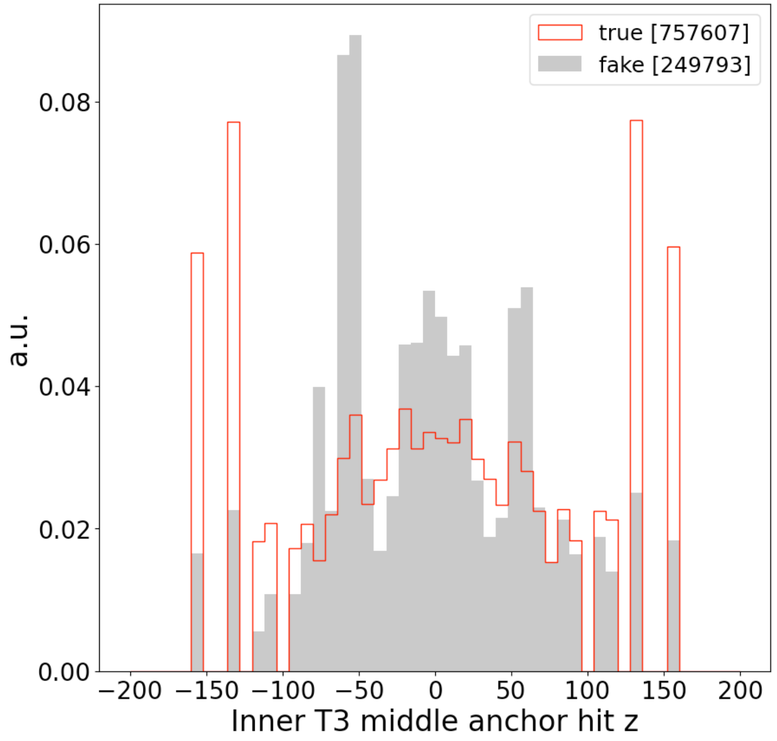


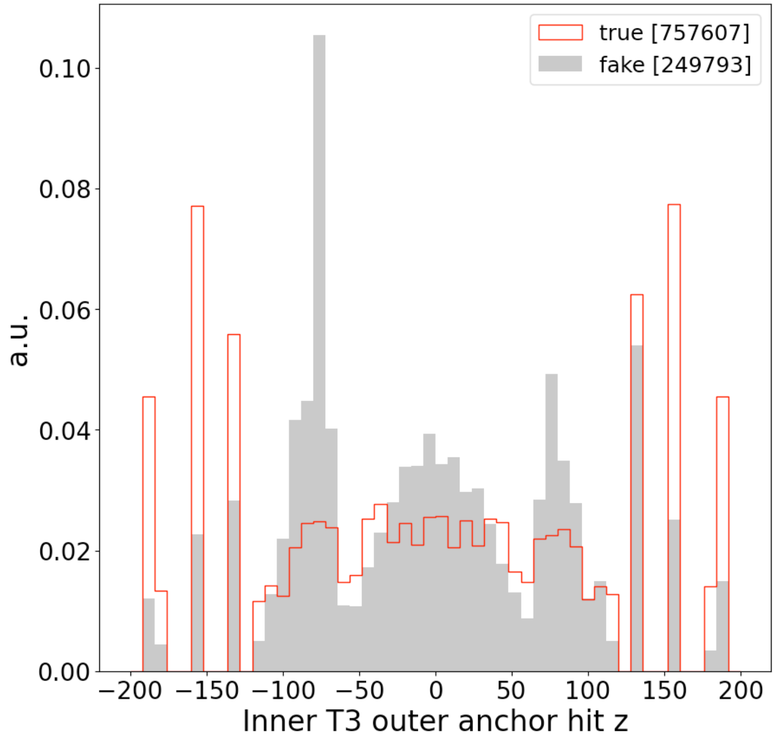
Trial 1: Input Features

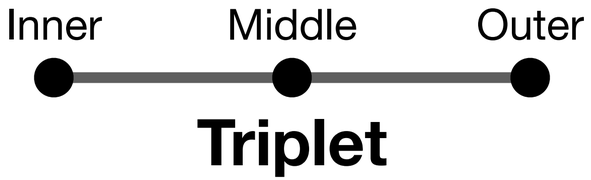
All plots available here





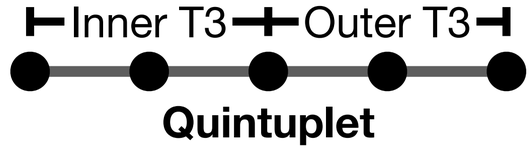


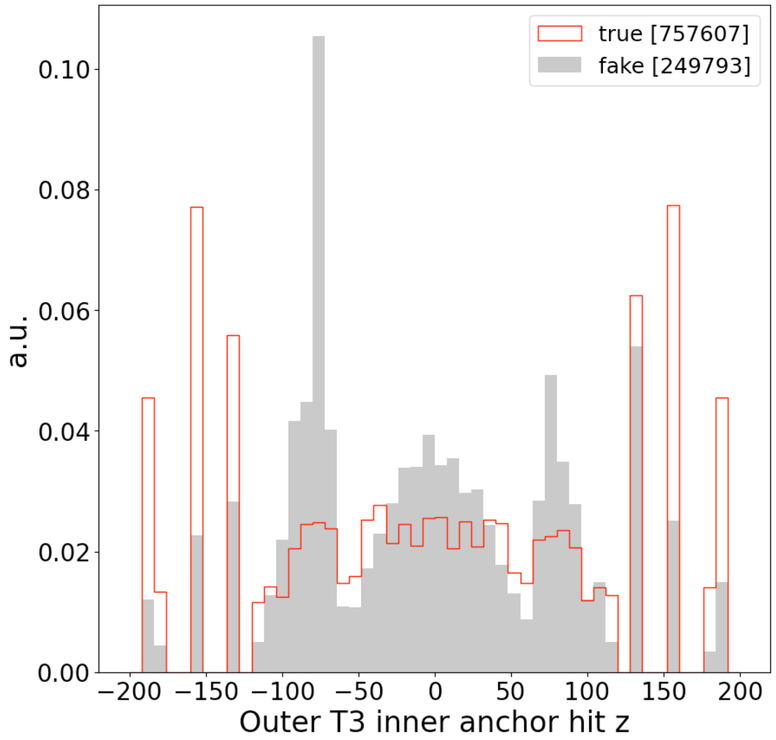


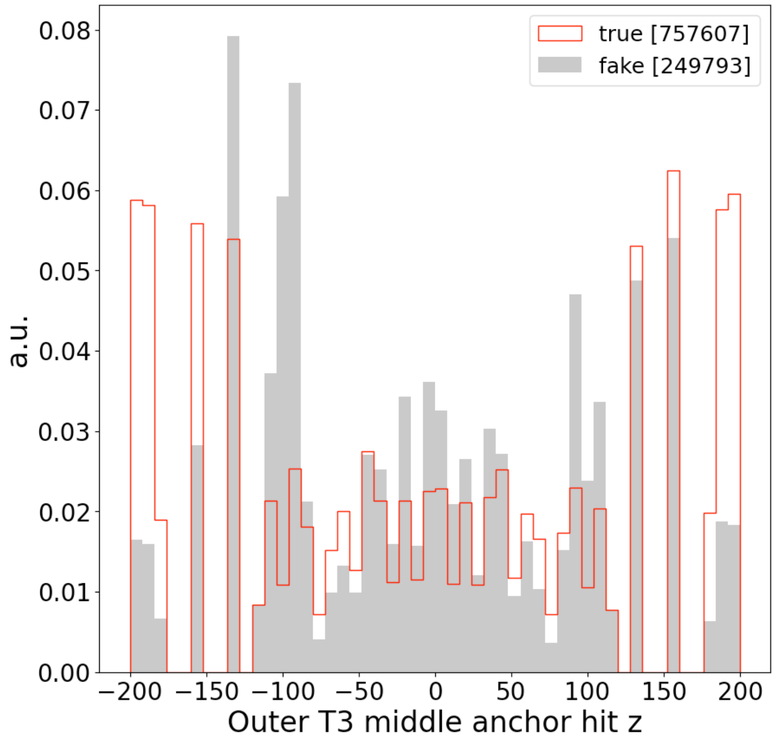


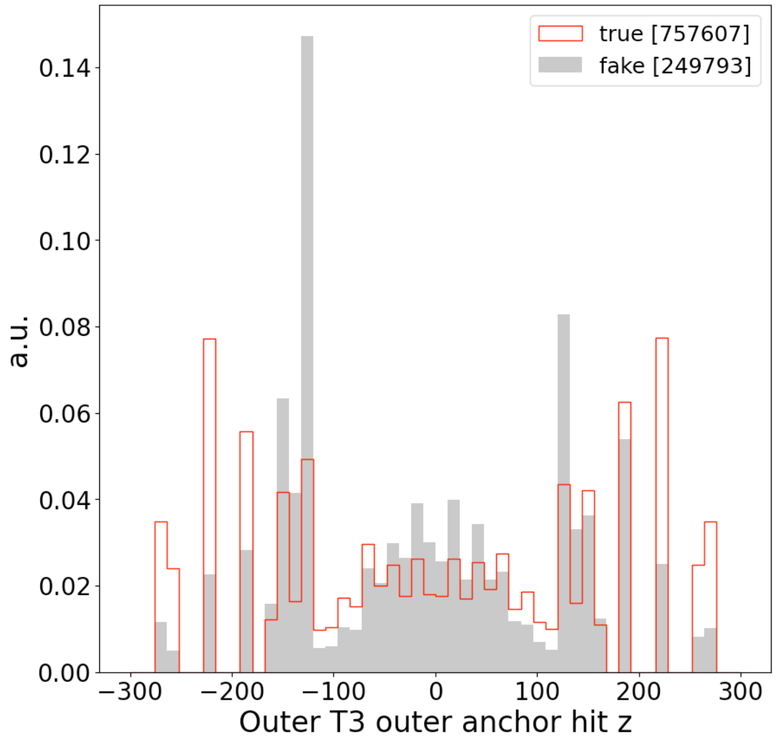
Trial 1: Input Features

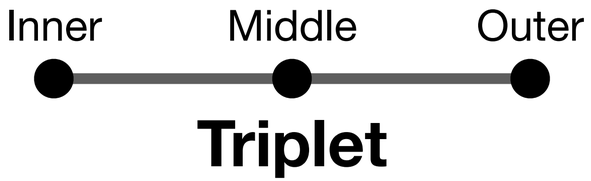
All plots available here





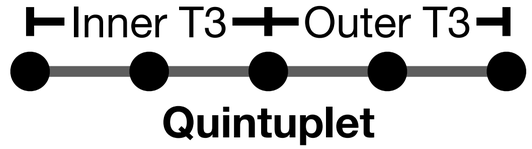


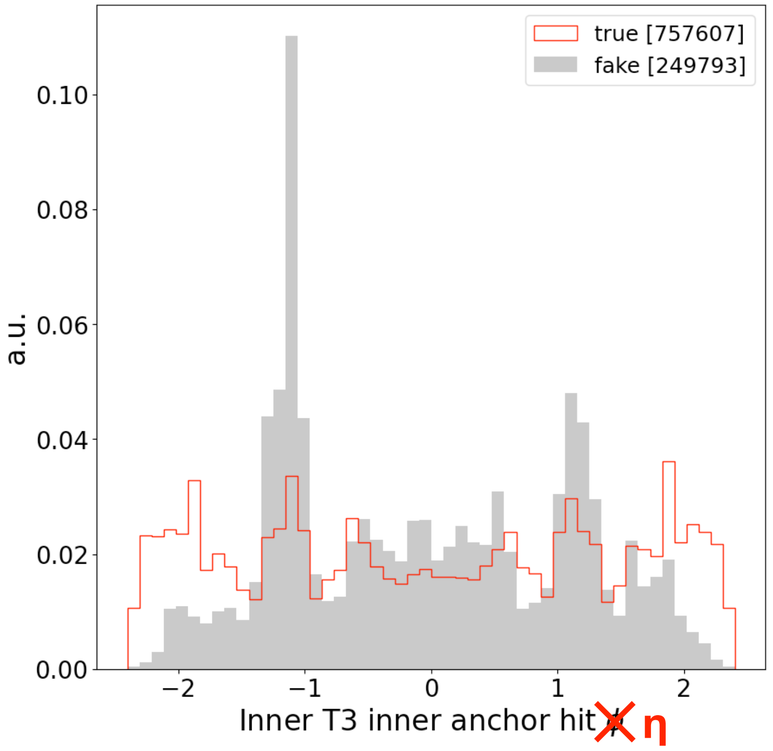


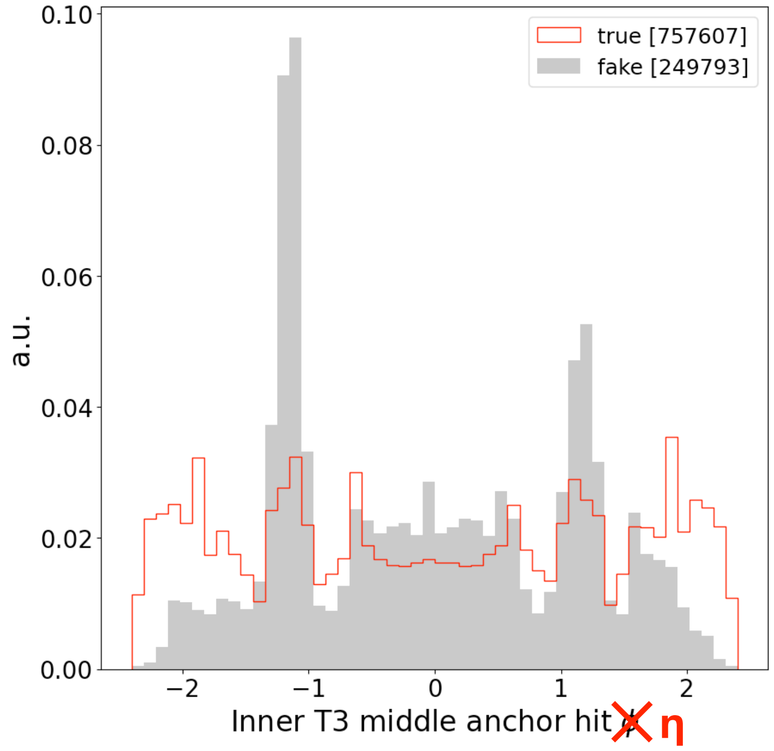


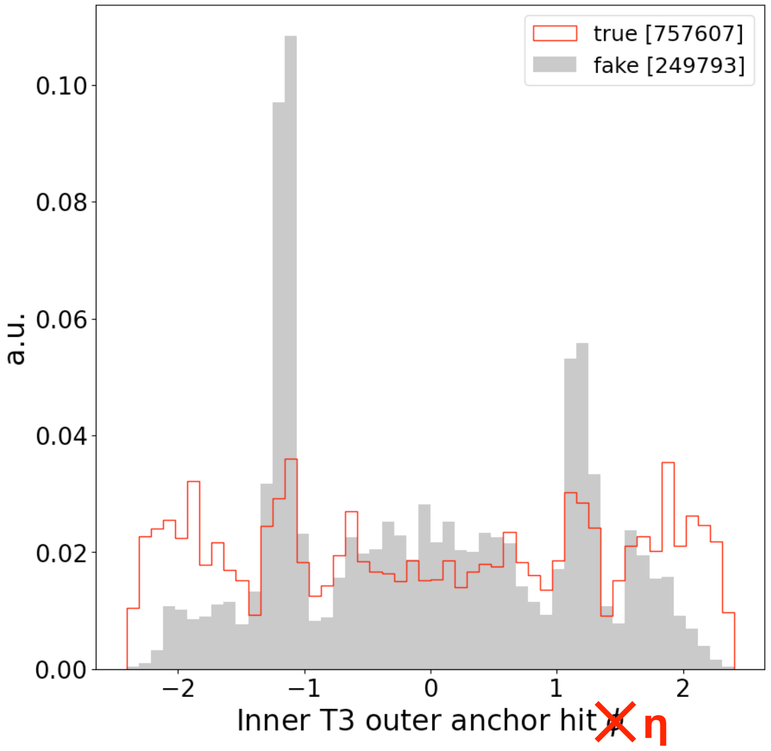
Trial 1: Input Features

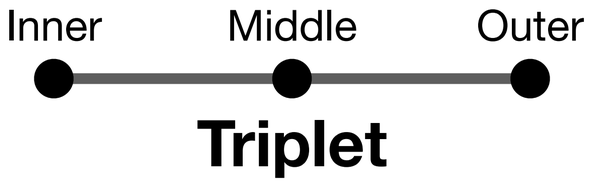
All plots available here





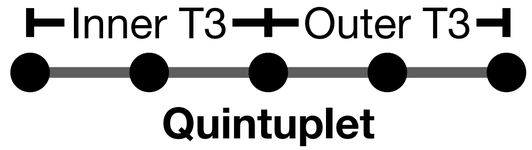


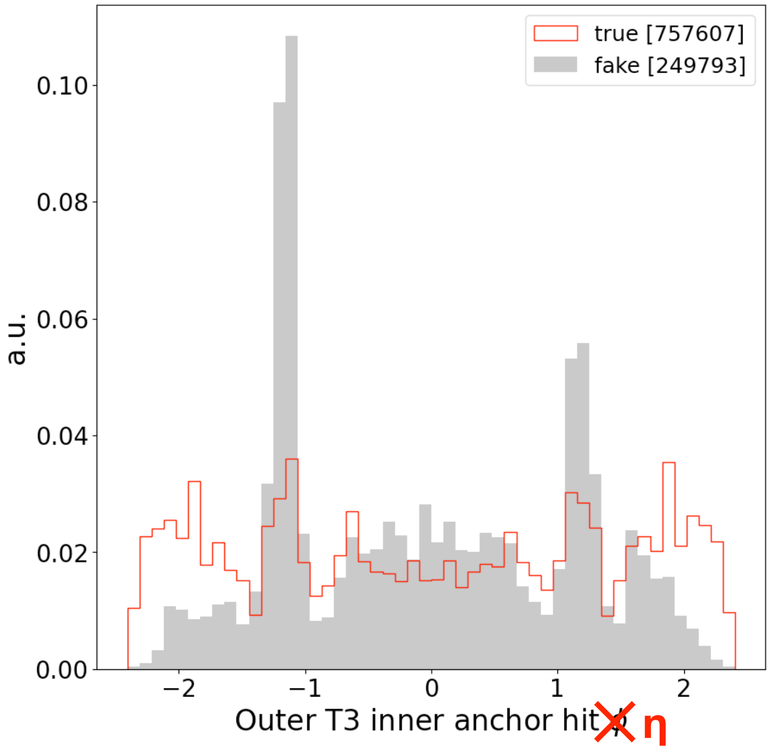


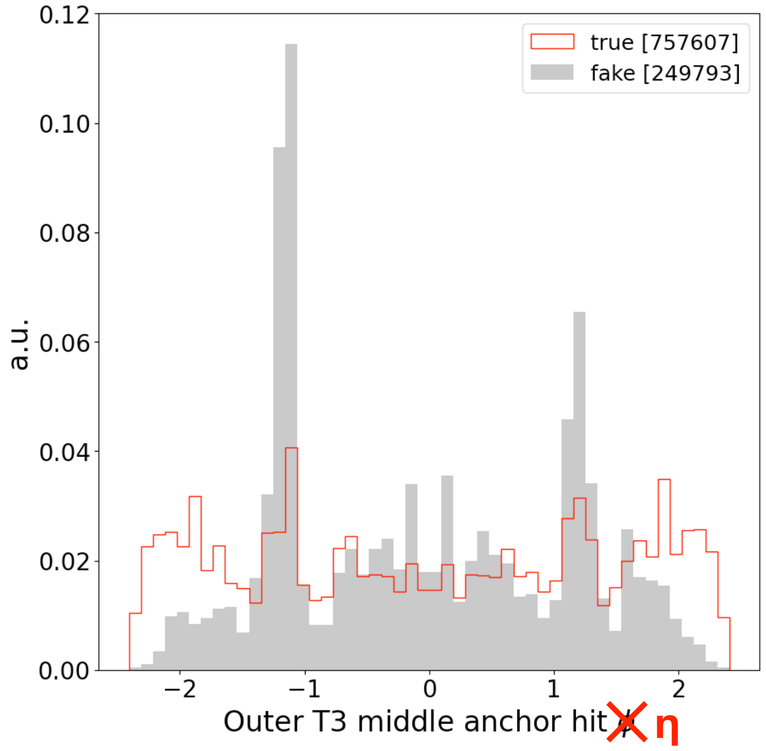


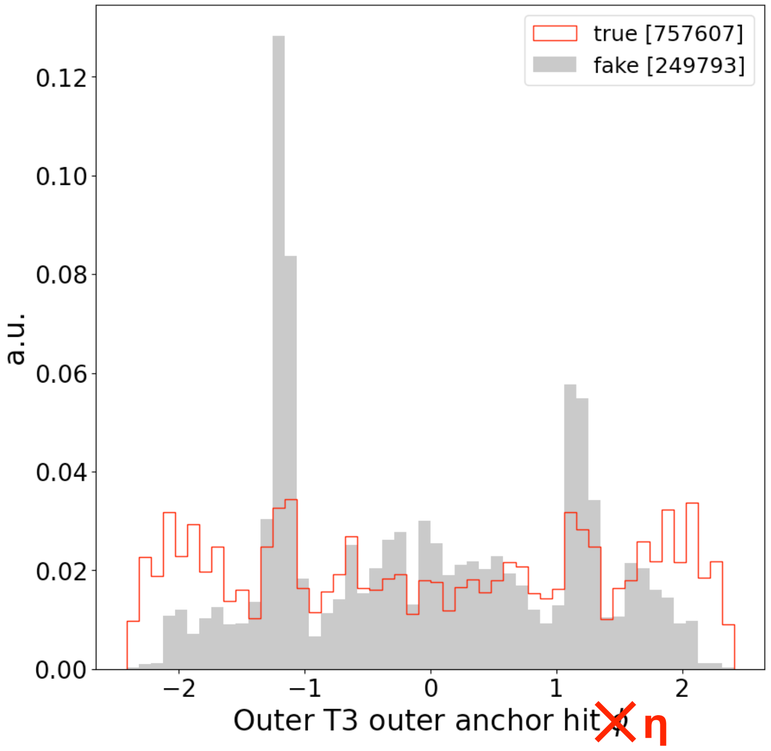
Trial 1: Input Features

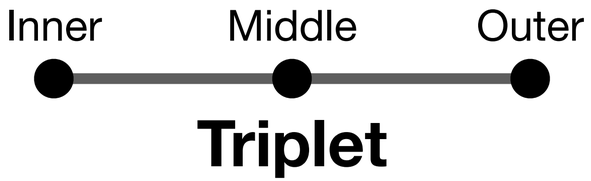
All plots available here





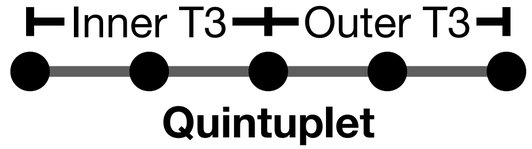


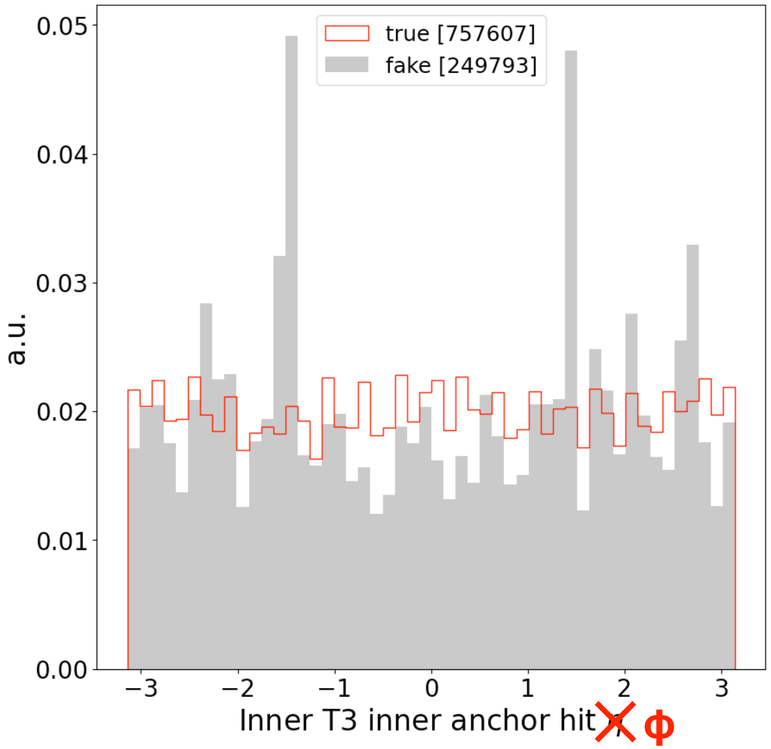


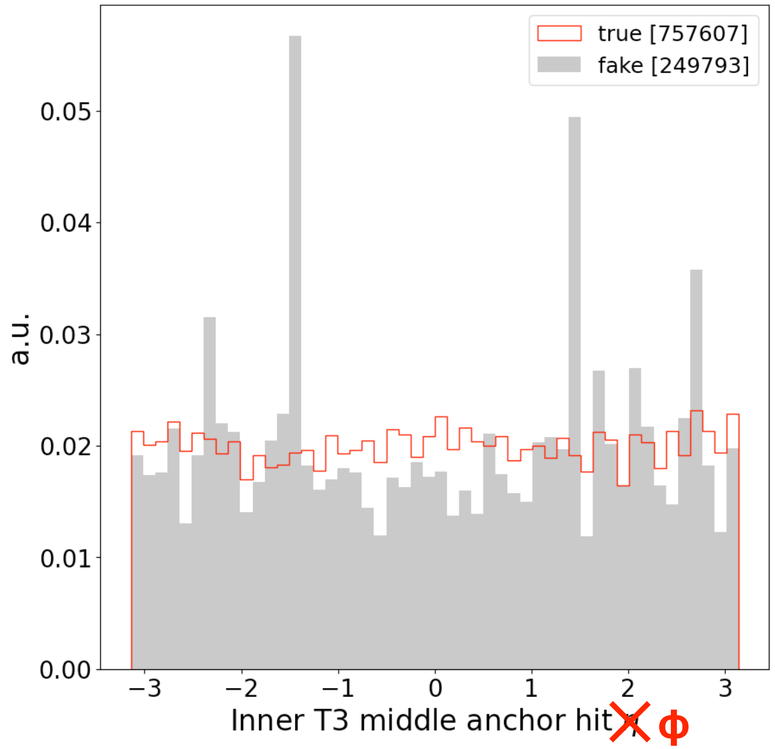


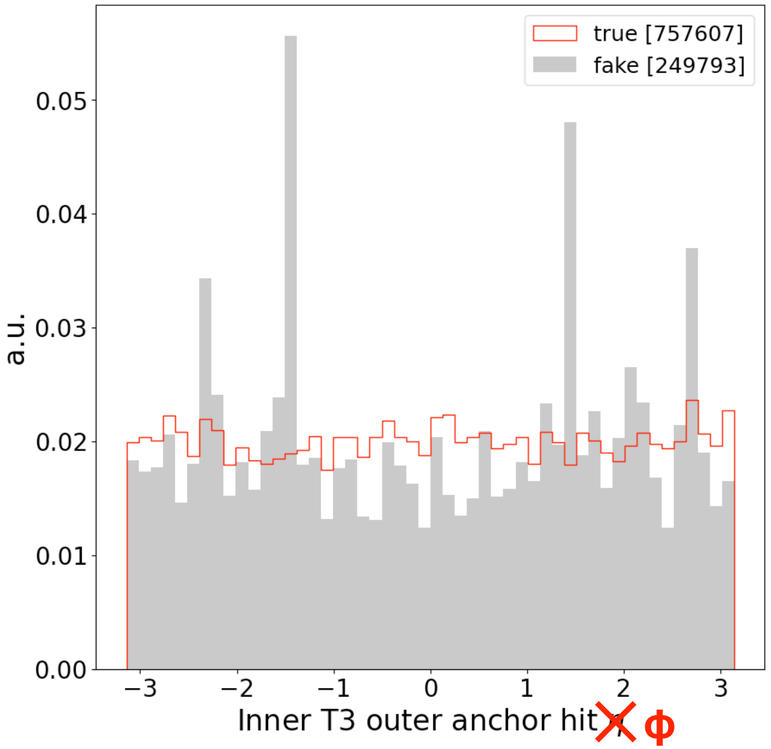
Trial 1: Input Features

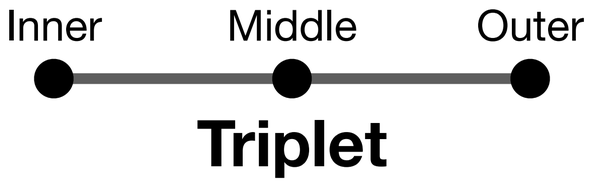
All plots available here





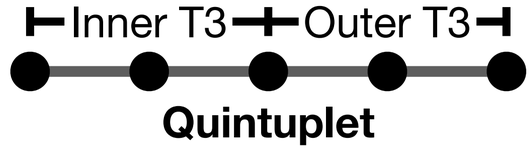


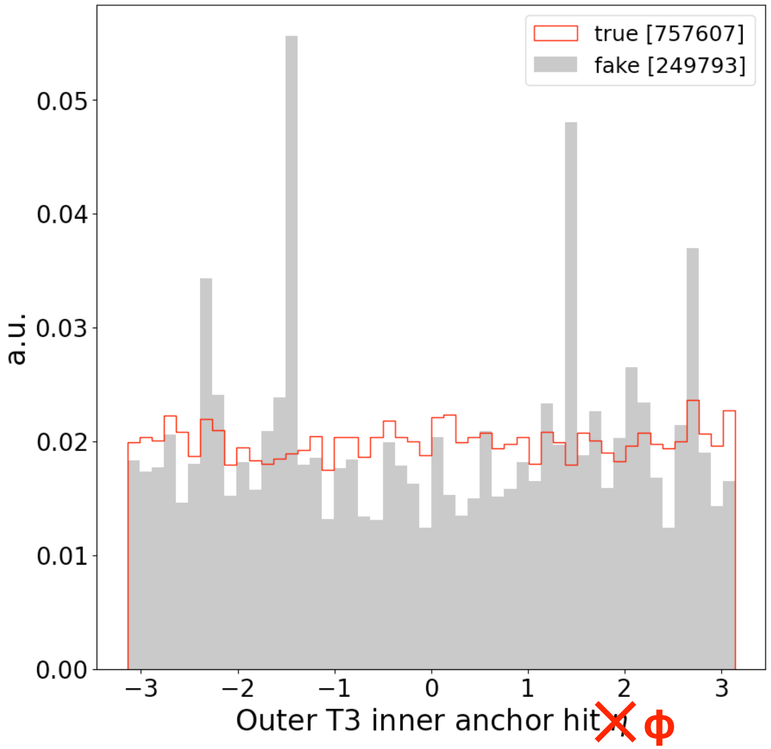


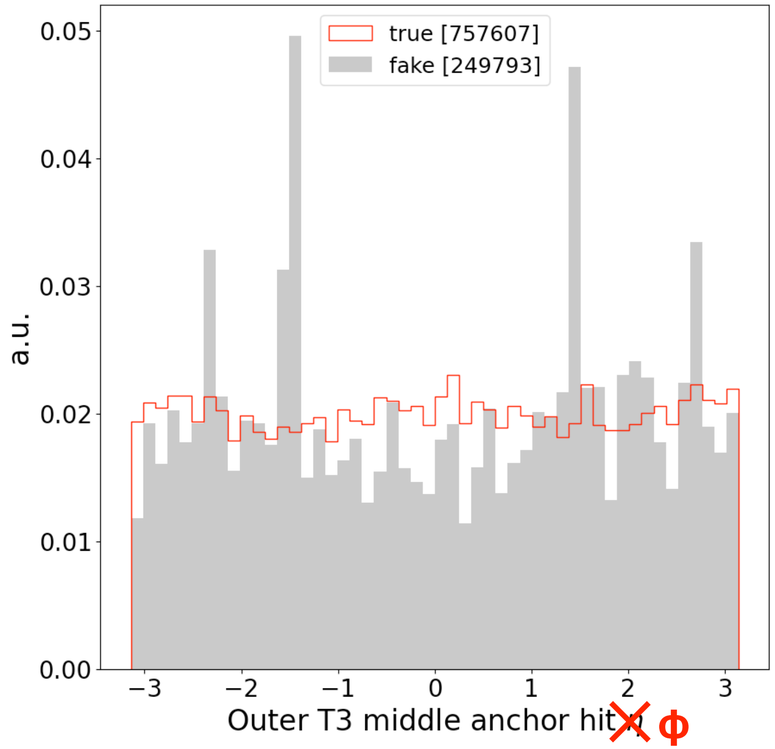


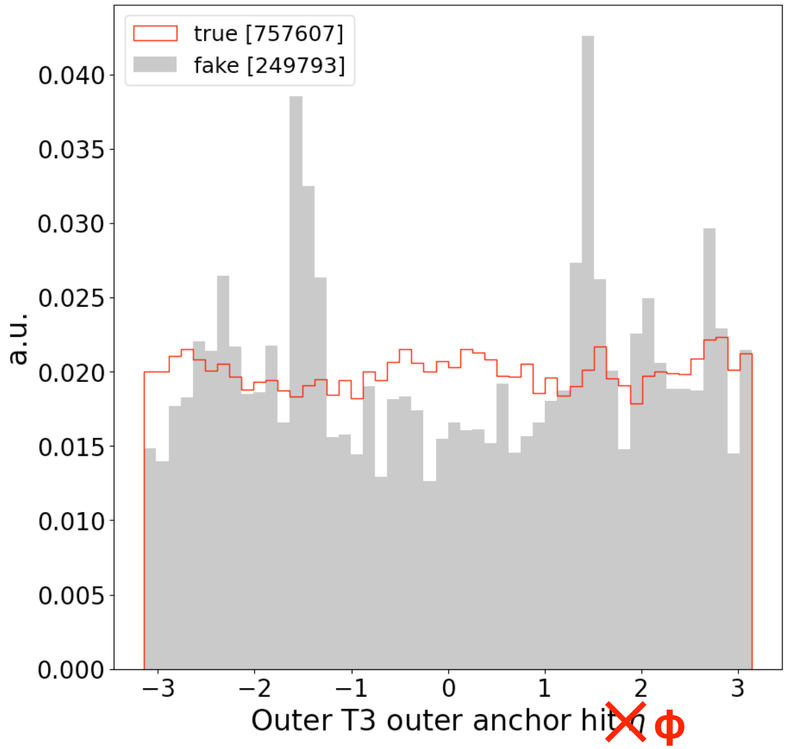
Trial 1: Input Features

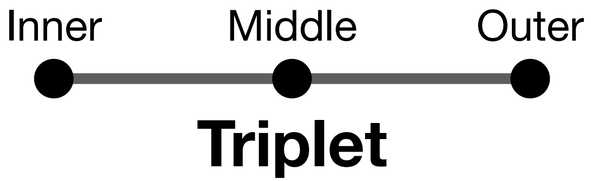
All plots available here





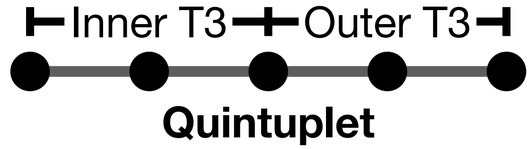


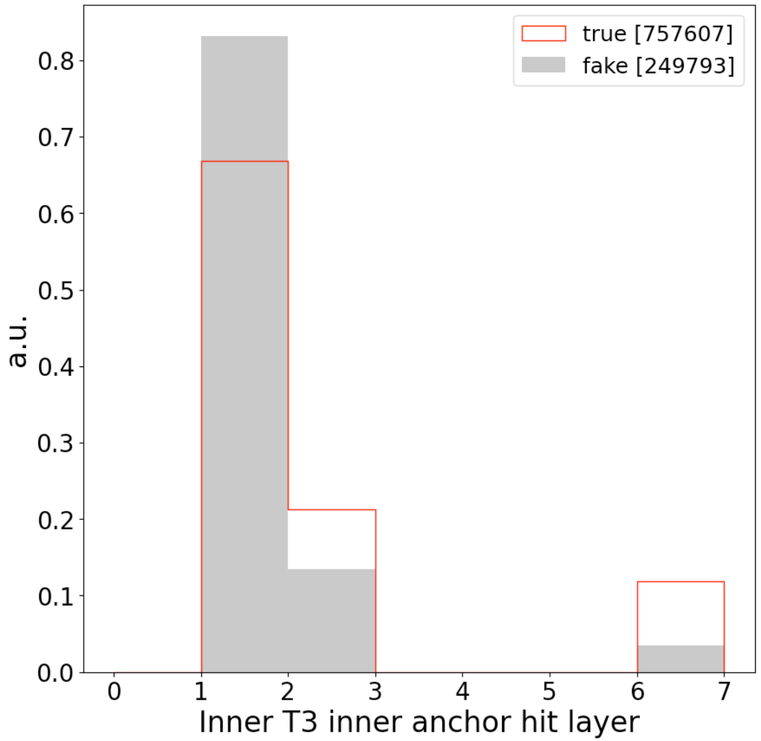


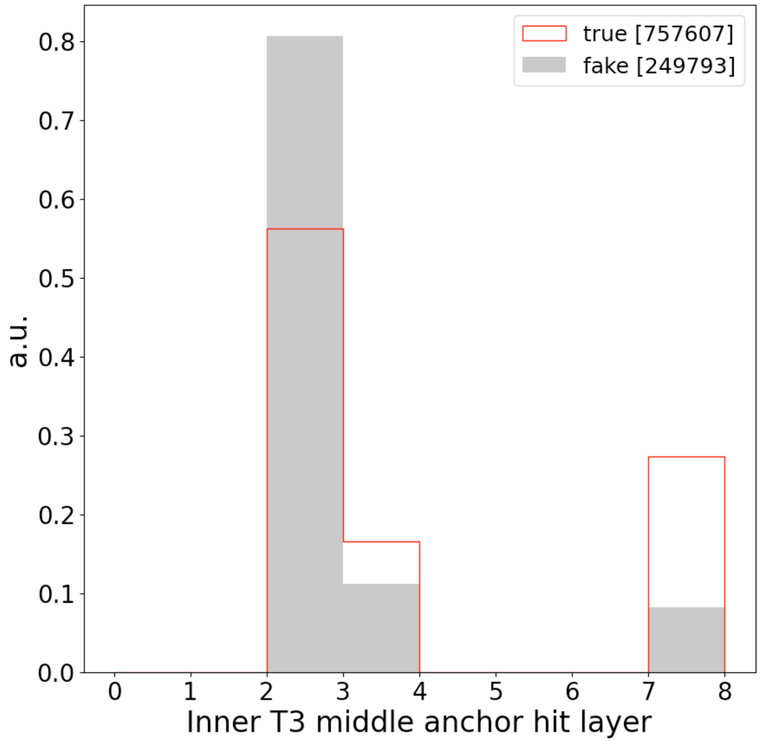


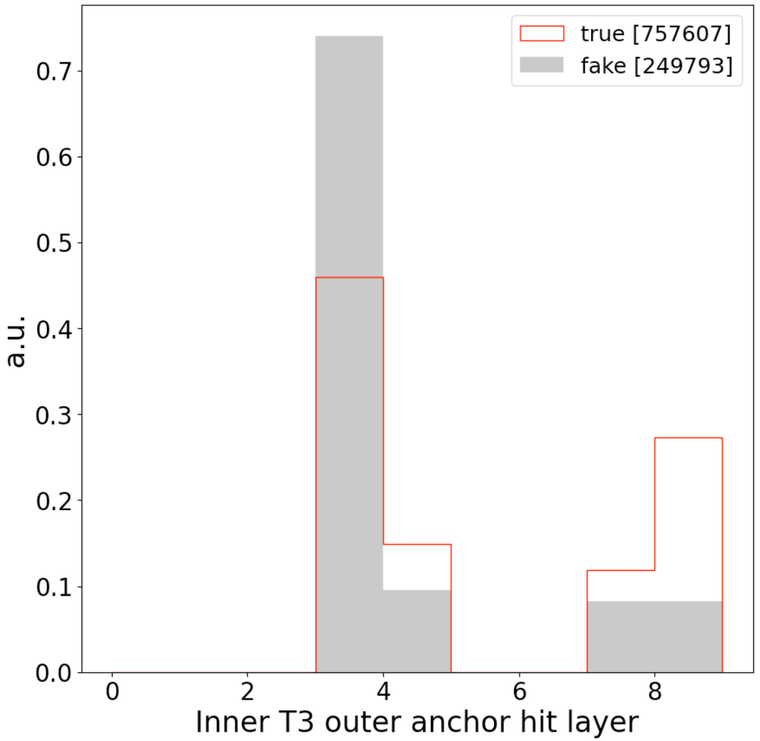
Trial 1: Input Features

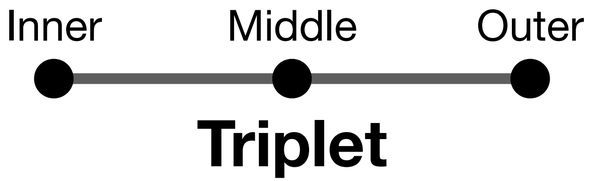
All plots available here





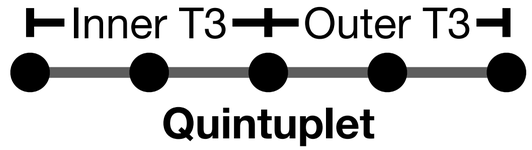


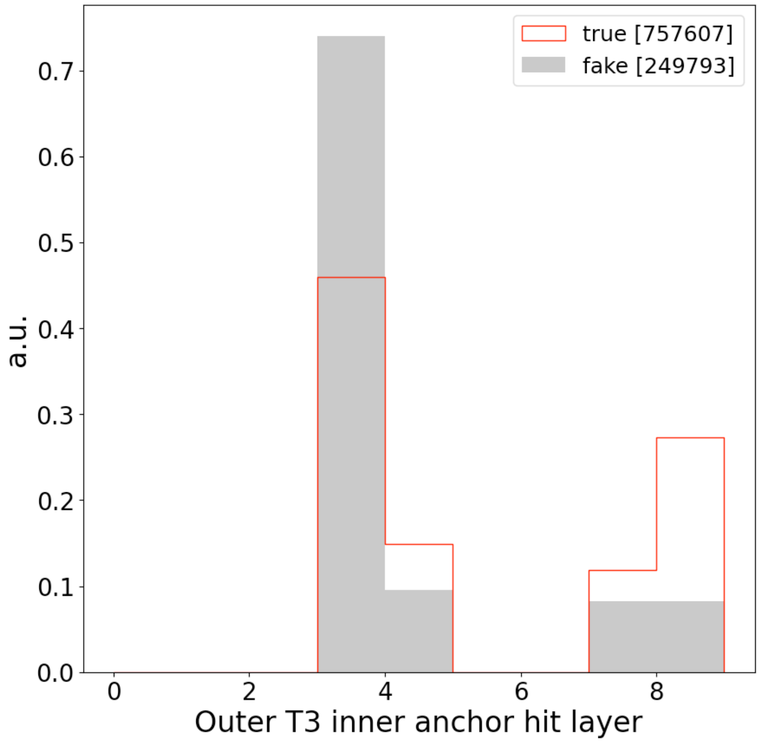


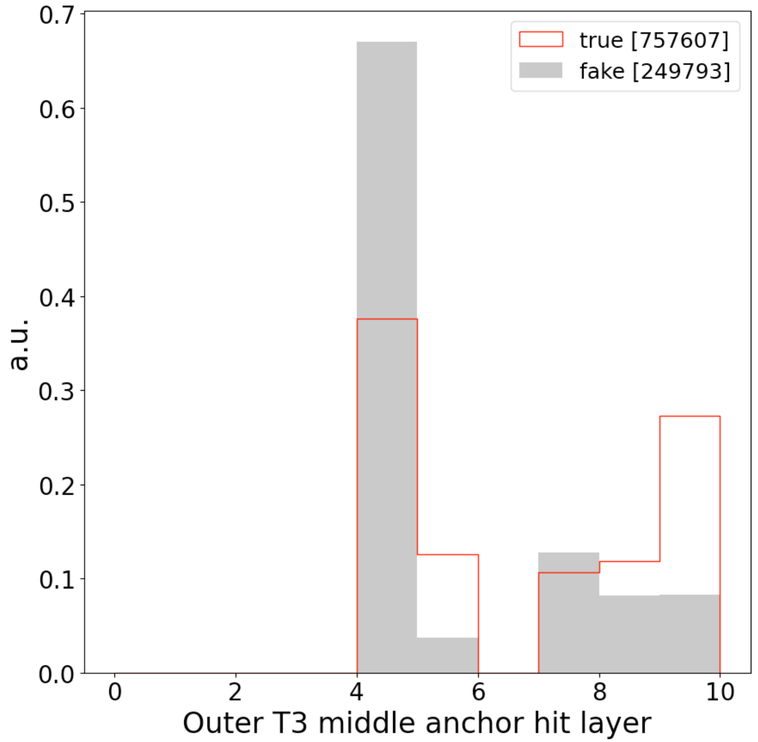


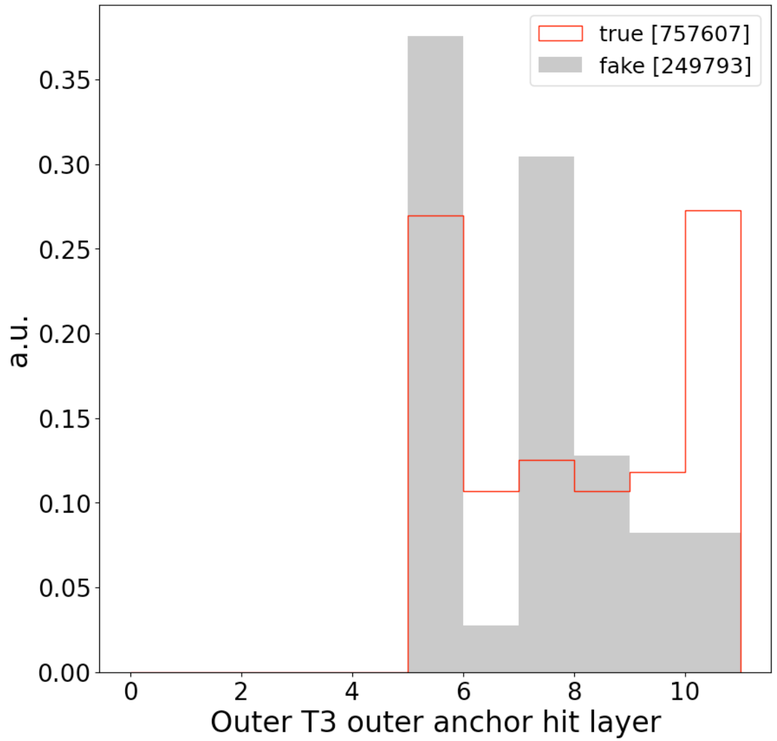
Trial 1: Input Features

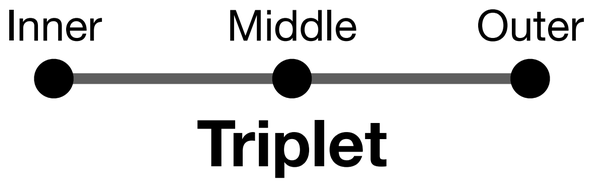
All plots available here







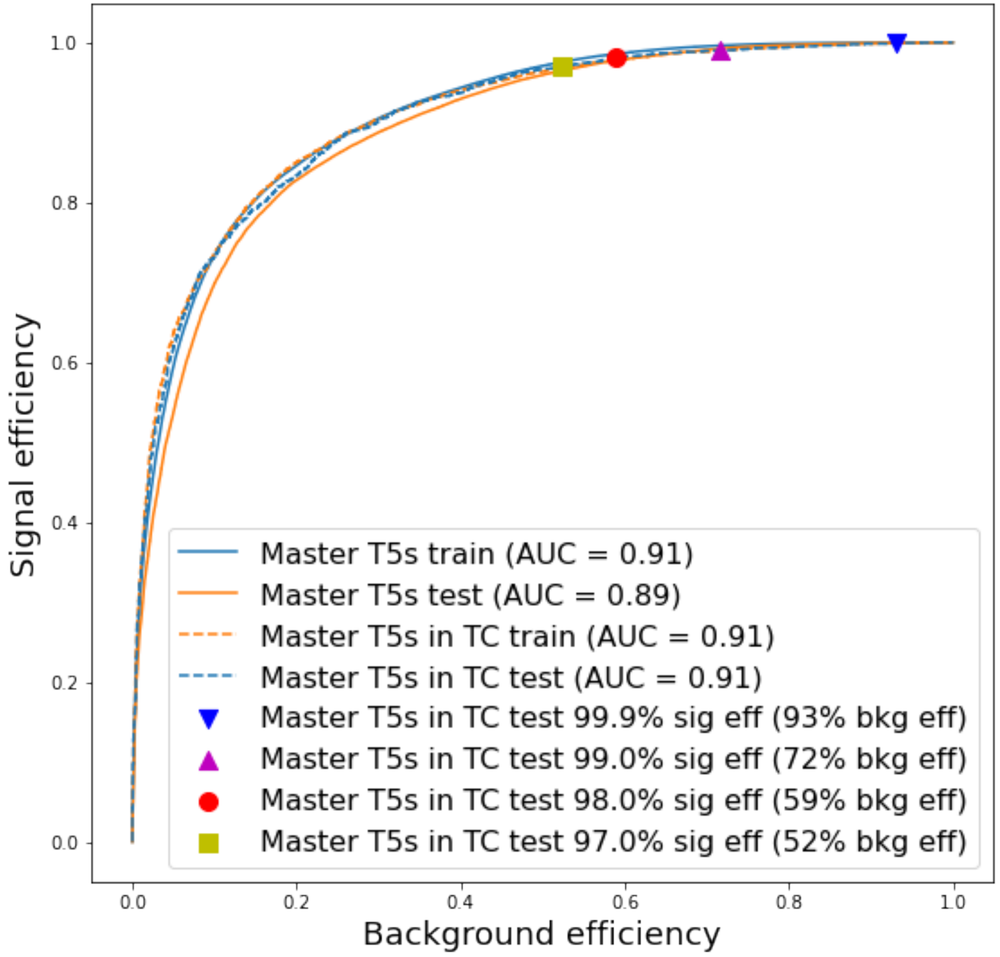




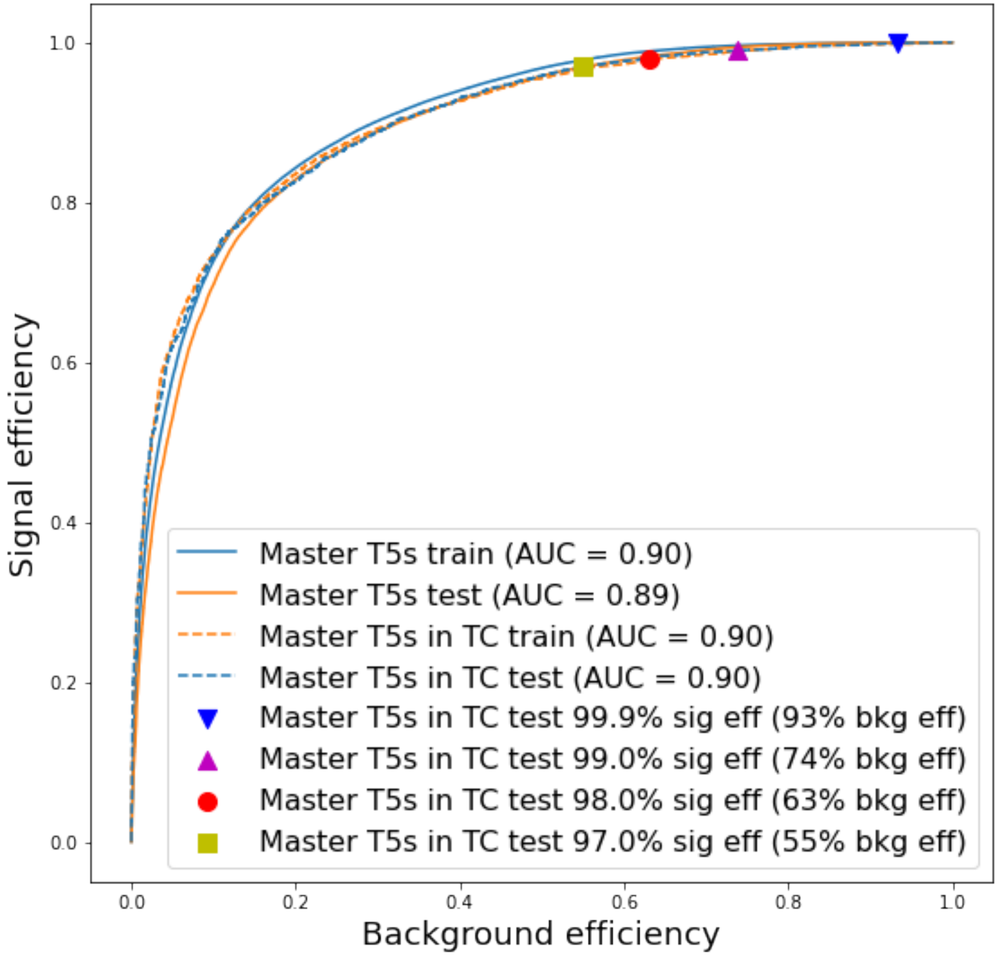
Trial 1: ~~GNN~~DNN Performance

Performance is comparable!

1 hidden layer

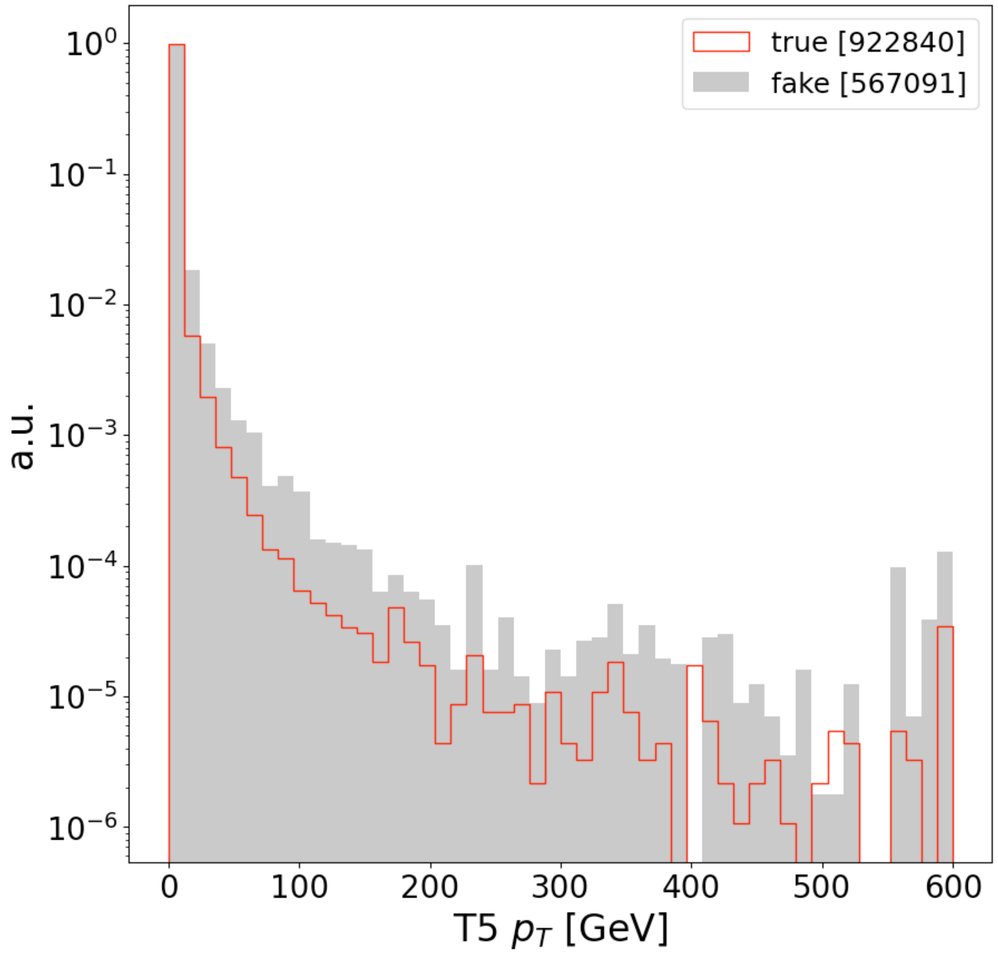


No hidden layers

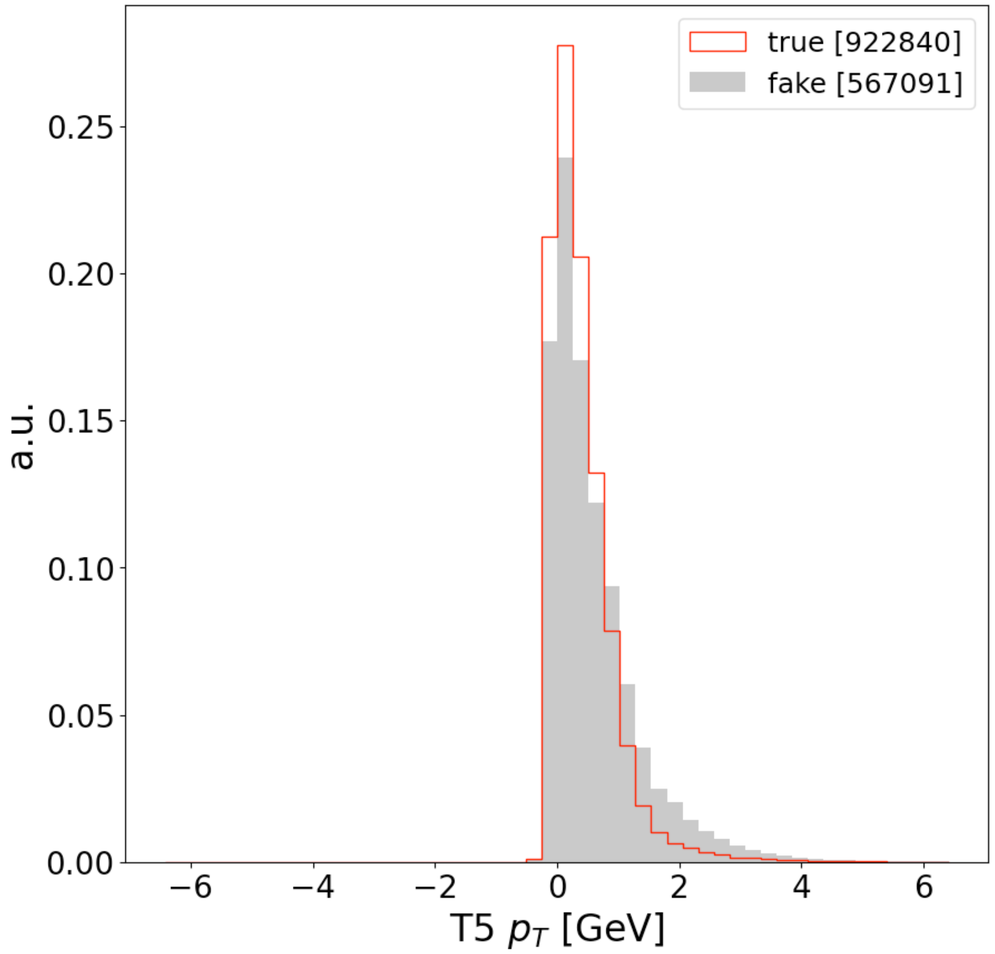


Trial 2: Input Features

All plots available here

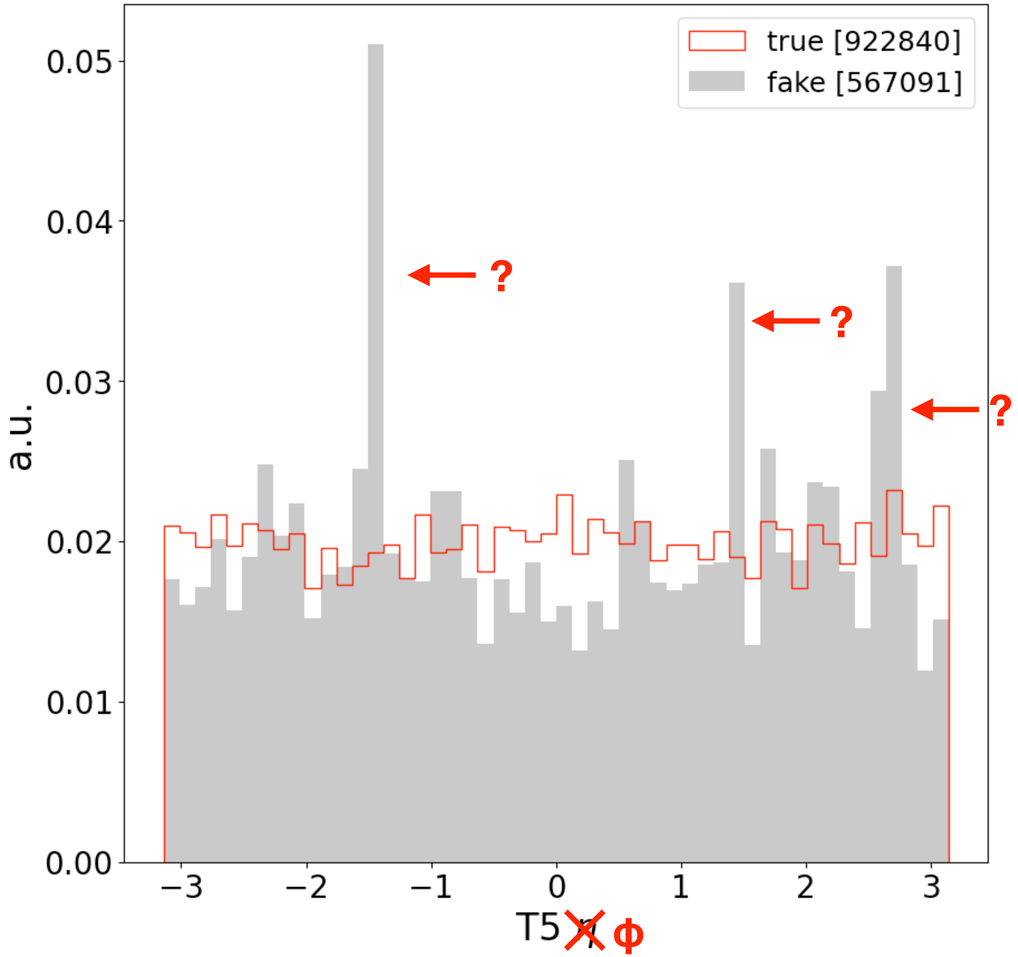


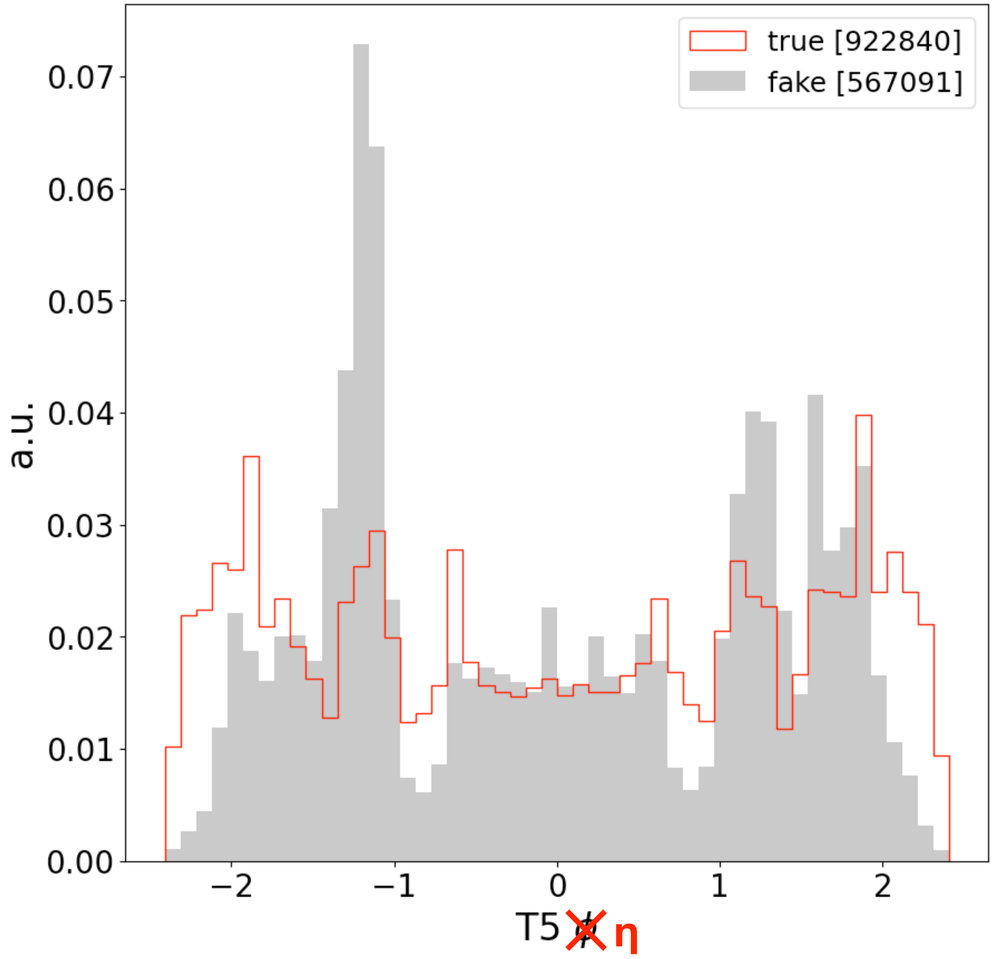
log(x)



Trial 2: Input Features

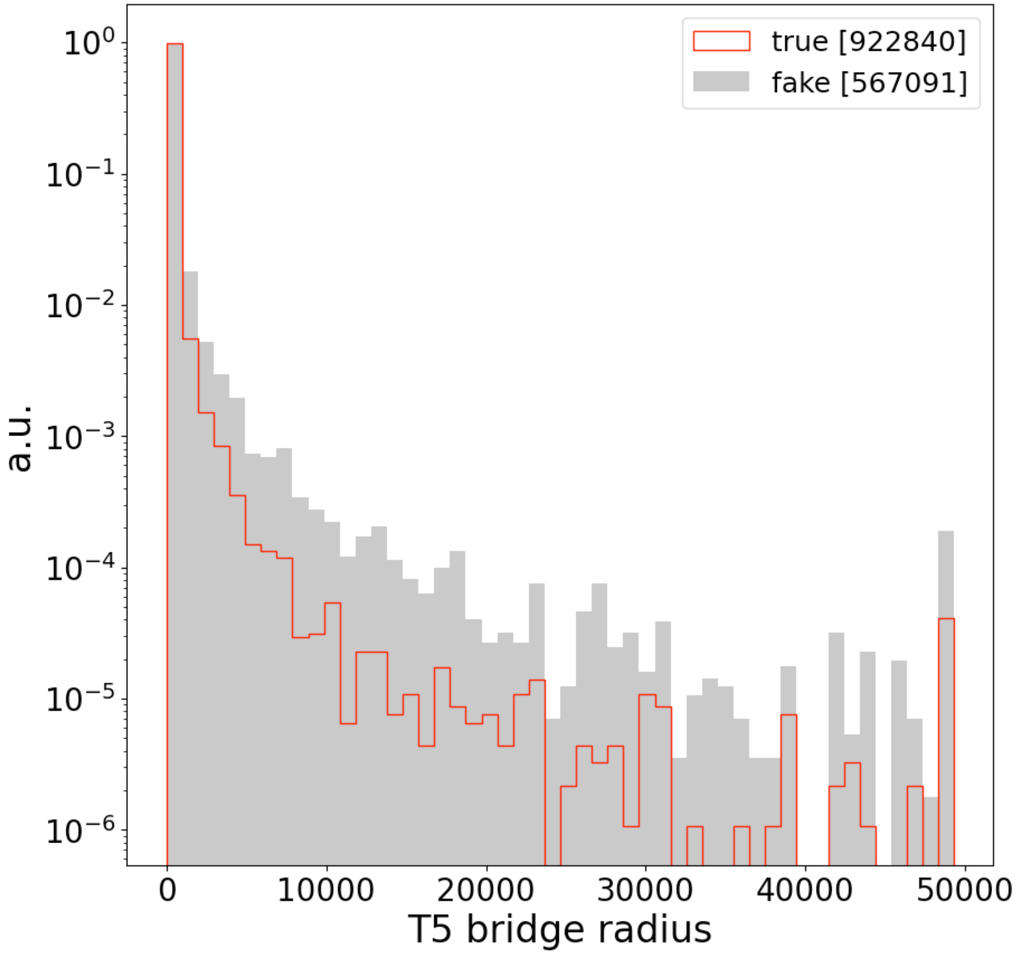
All plots available here



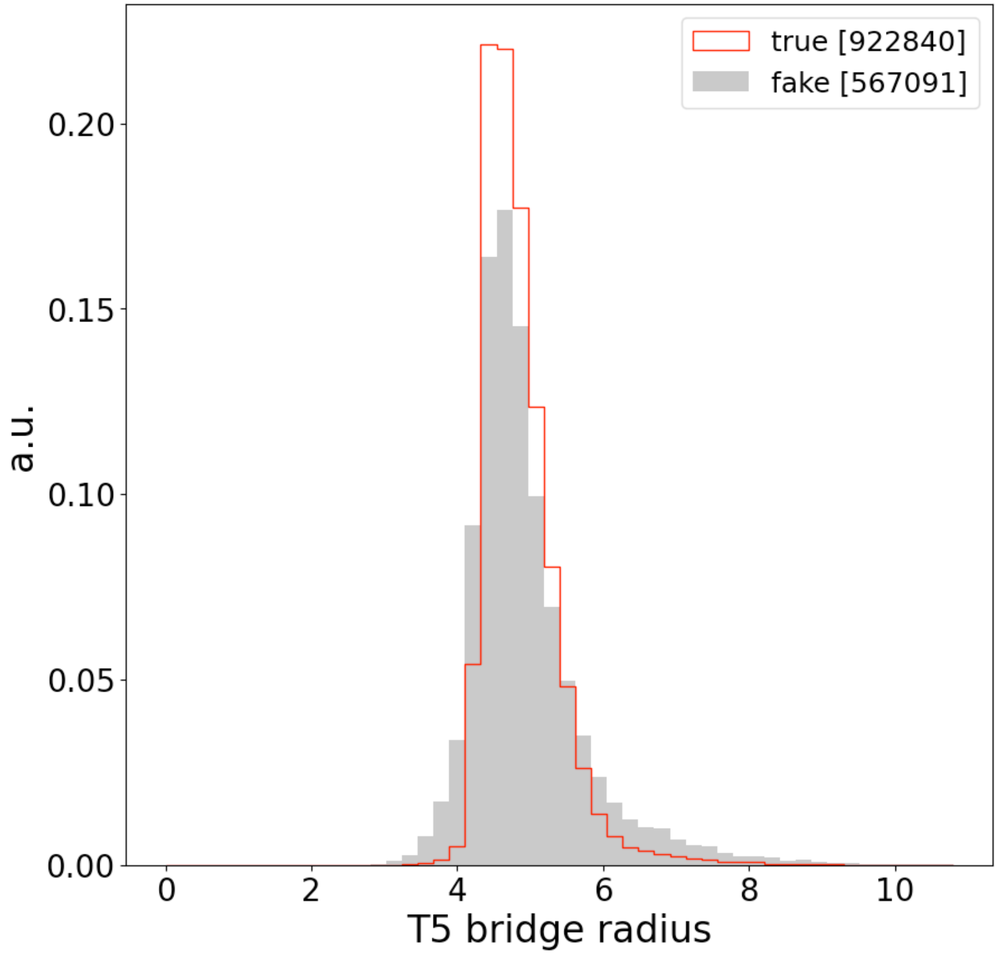


Trial 2: Input Features

All plots available here

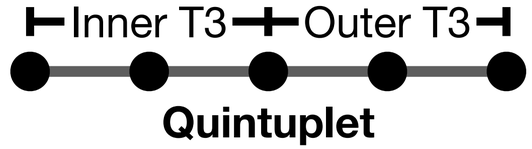


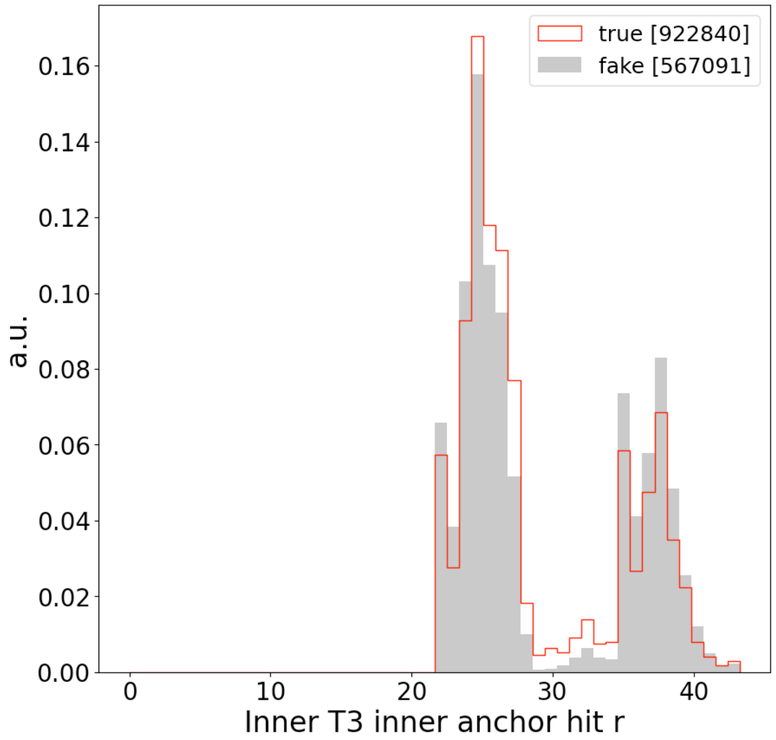
log(x)

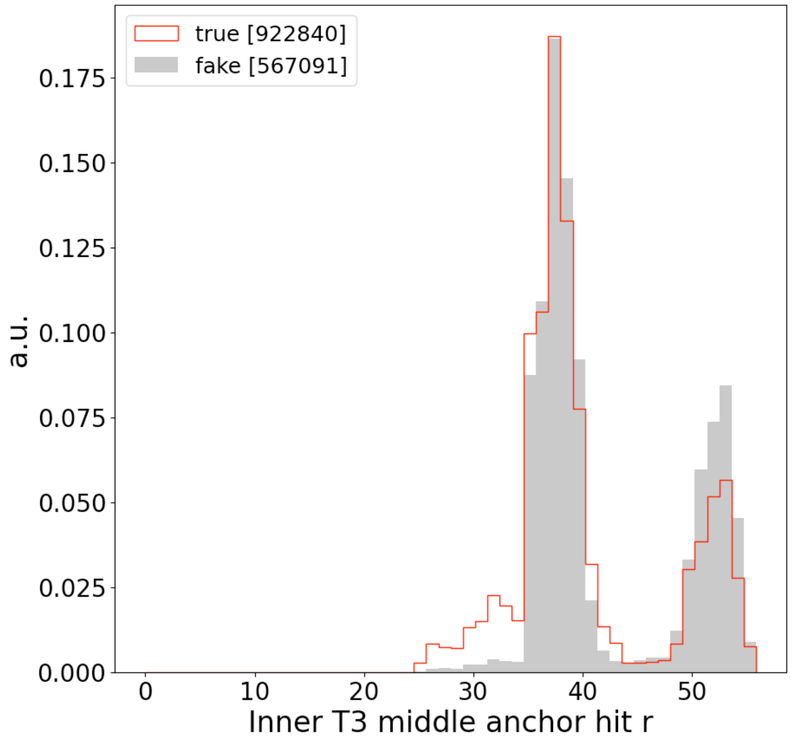


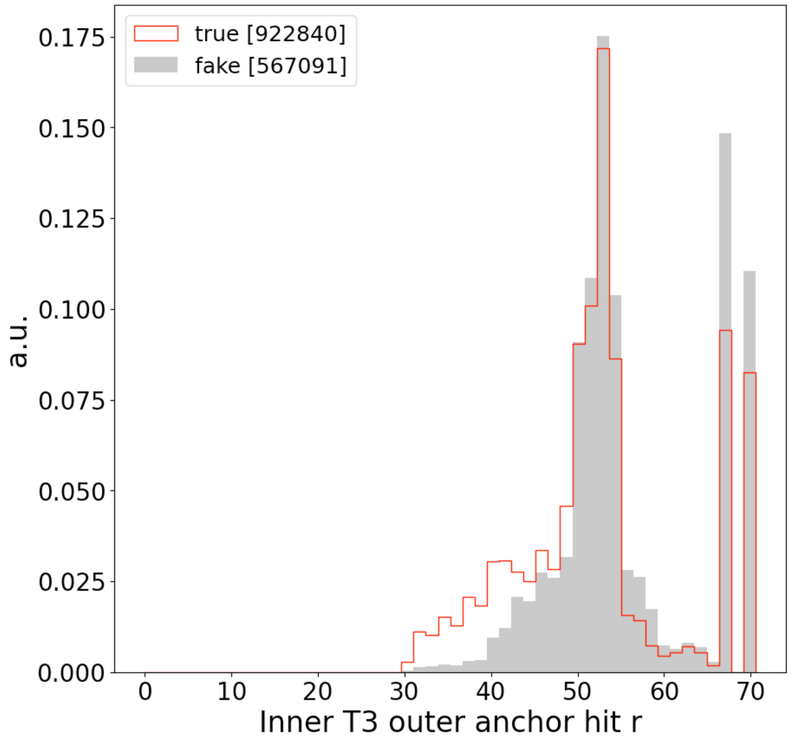
Trial 2: Input Features

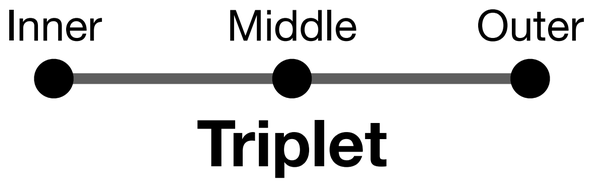
All plots available here





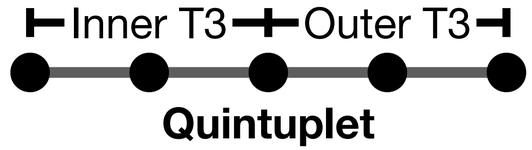


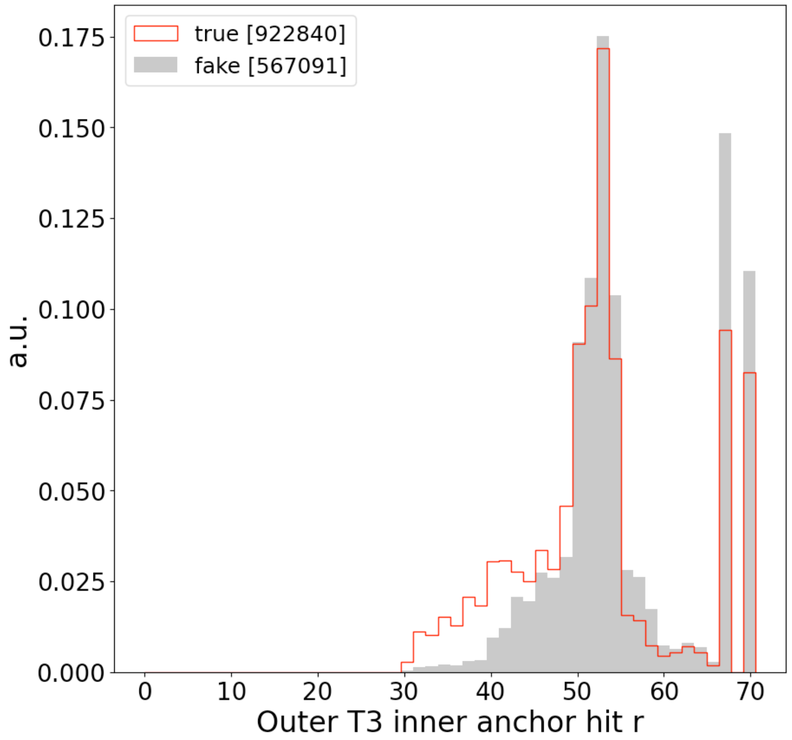


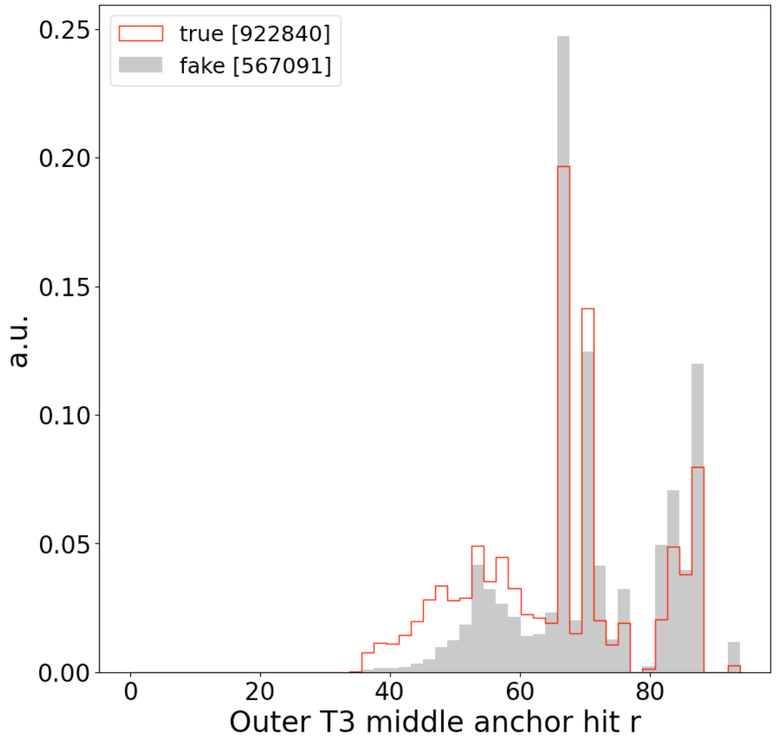


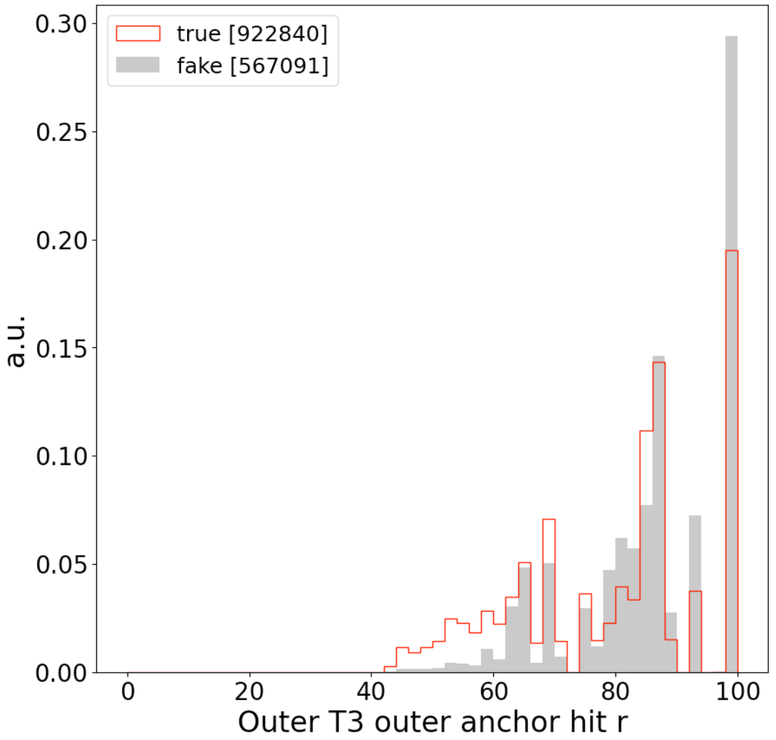
Trial 2: Input Features

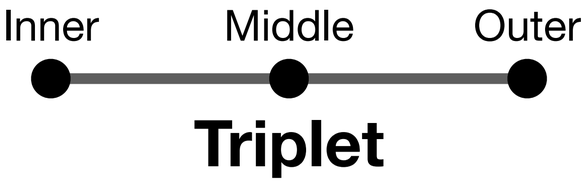
All plots available here





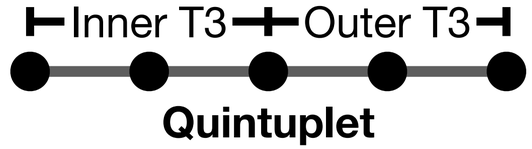


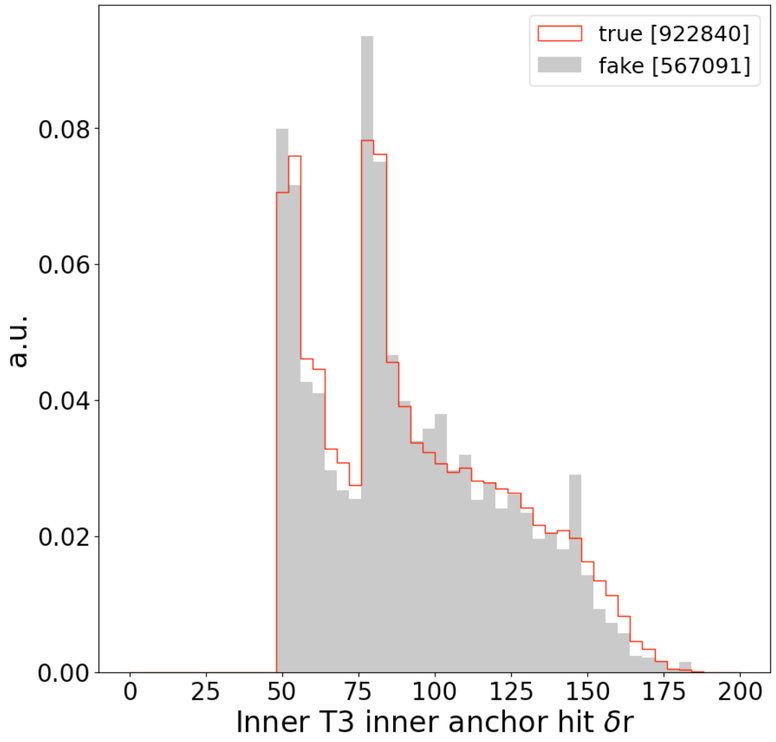


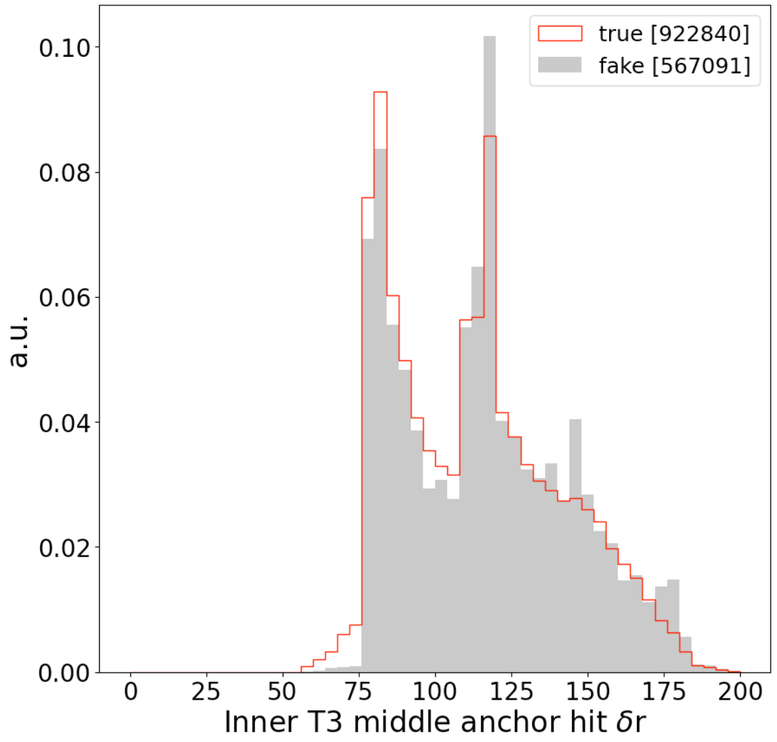


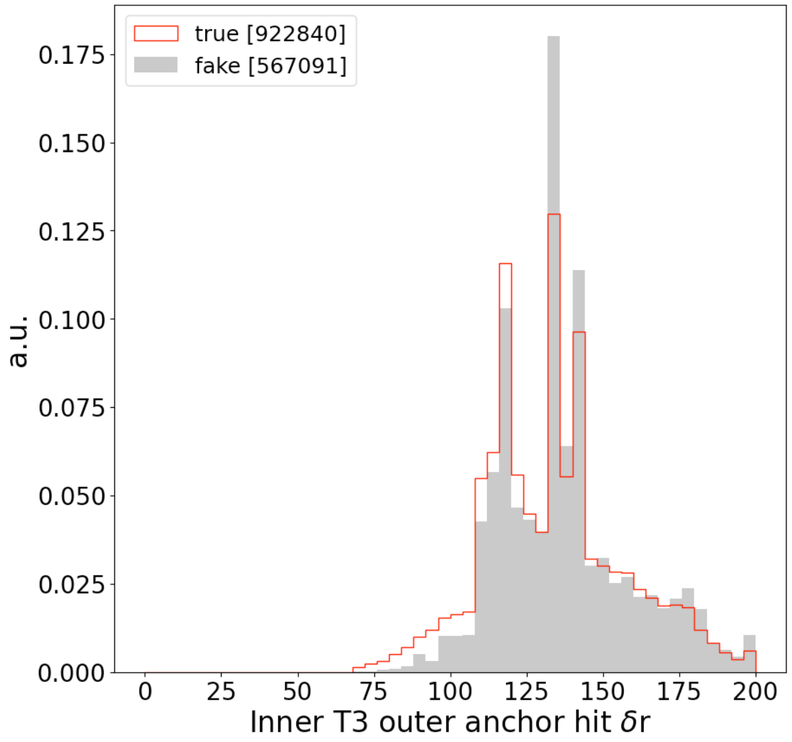
Trial 2: Input Features

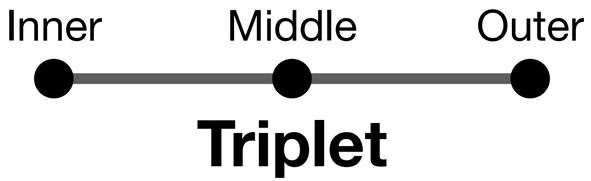
All plots available here





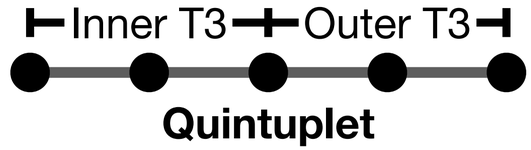


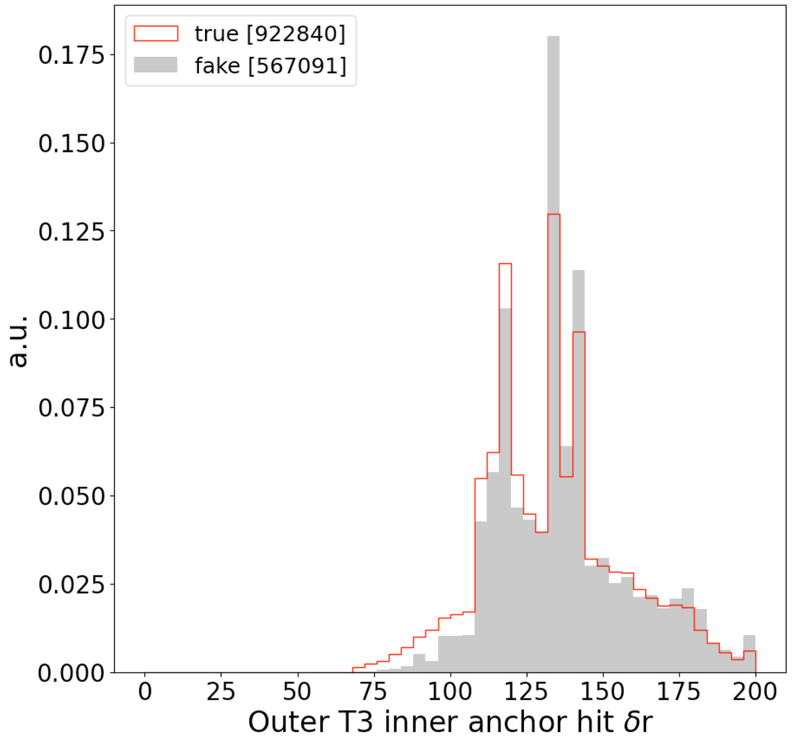


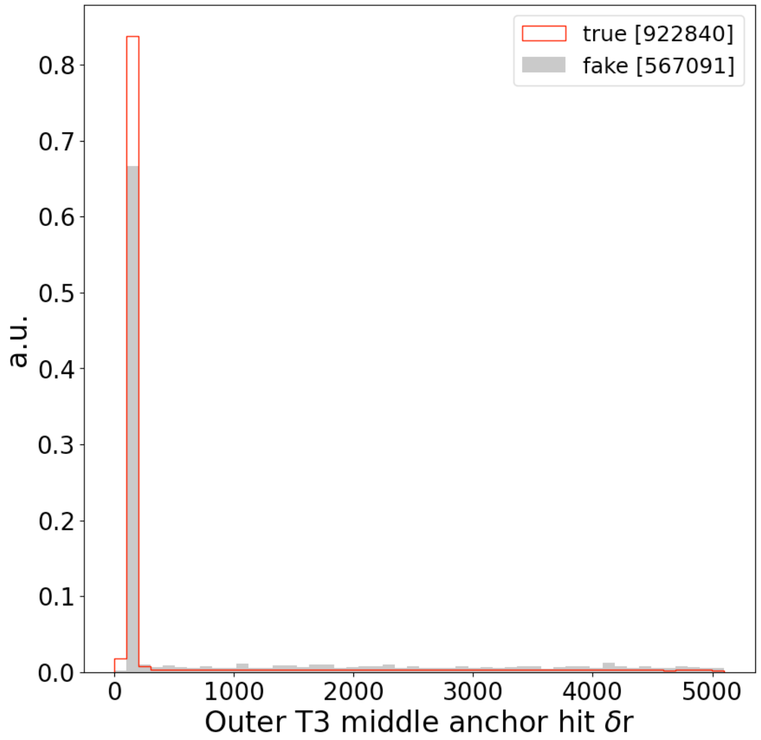


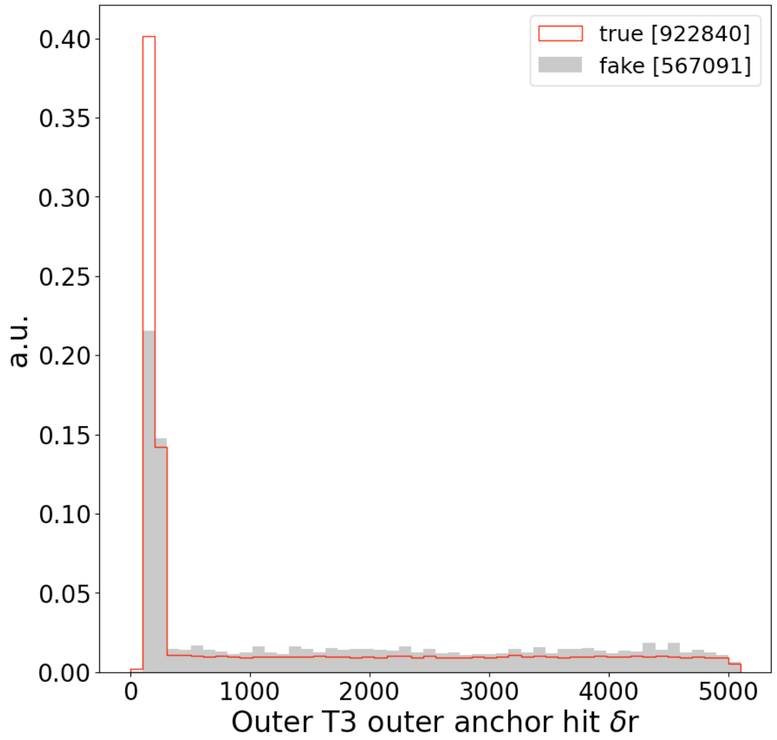
Trial 2: Input Features

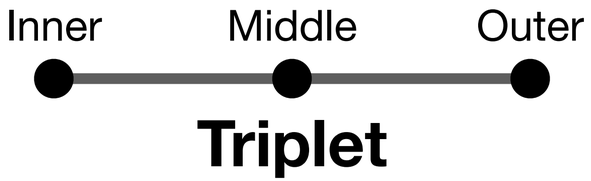
All plots available here





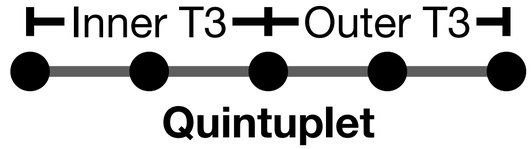


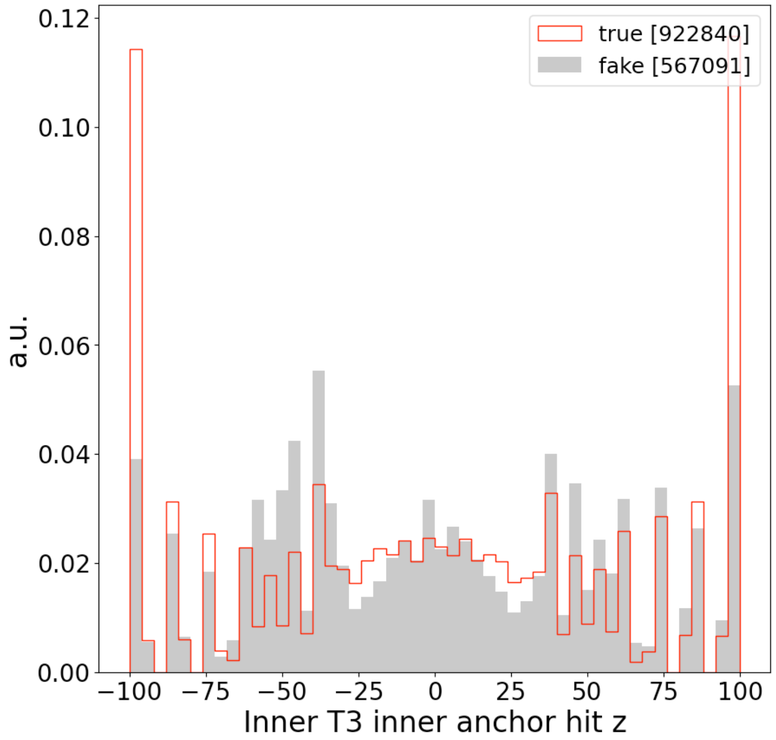


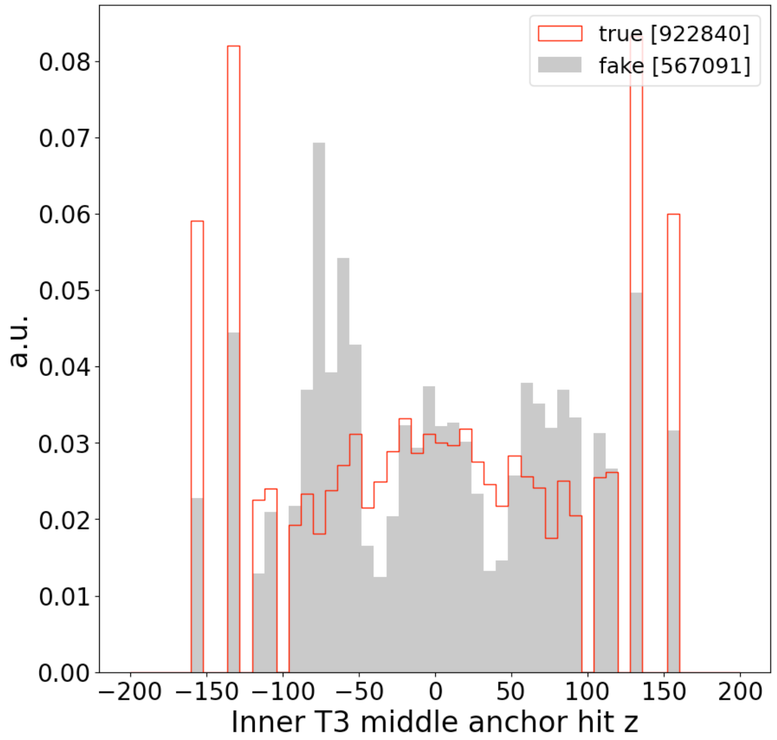


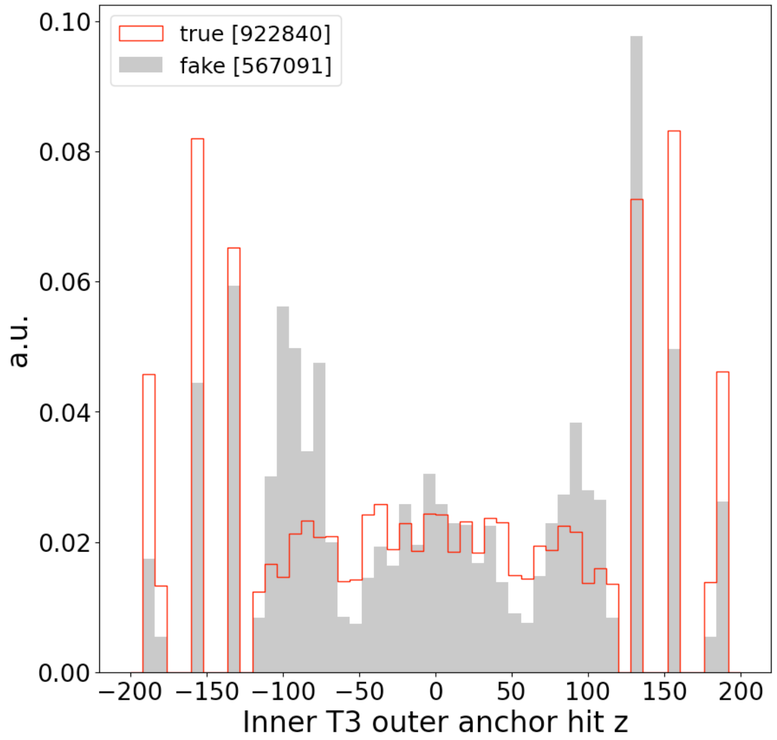
Trial 2: Input Features

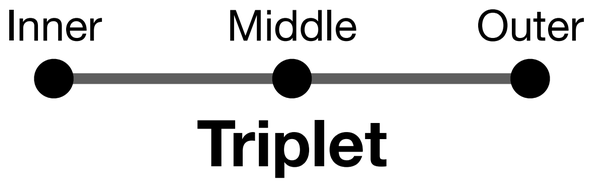
All plots available here





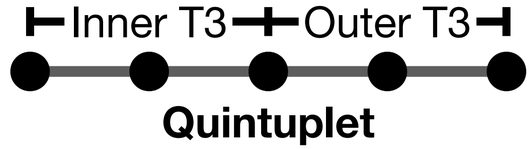


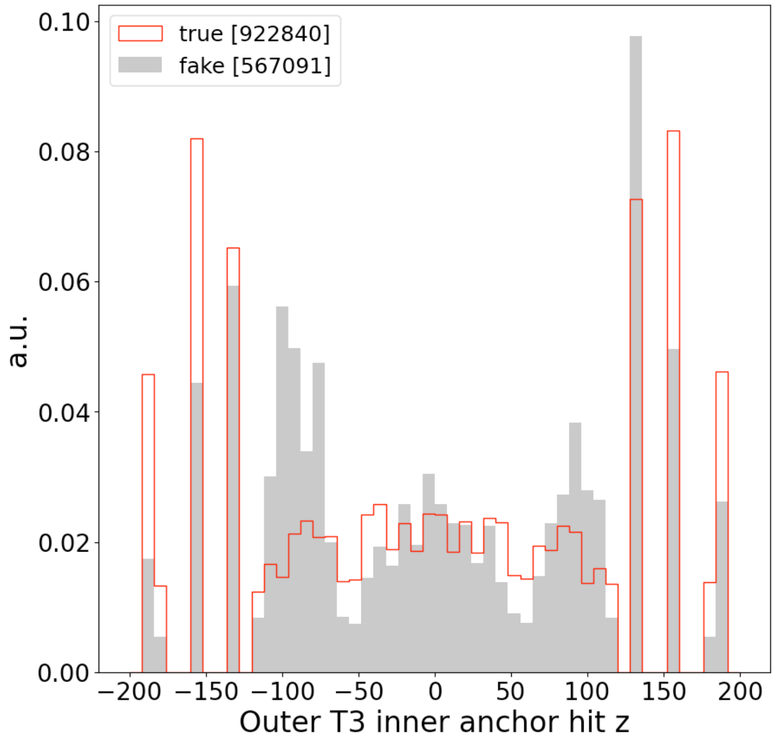


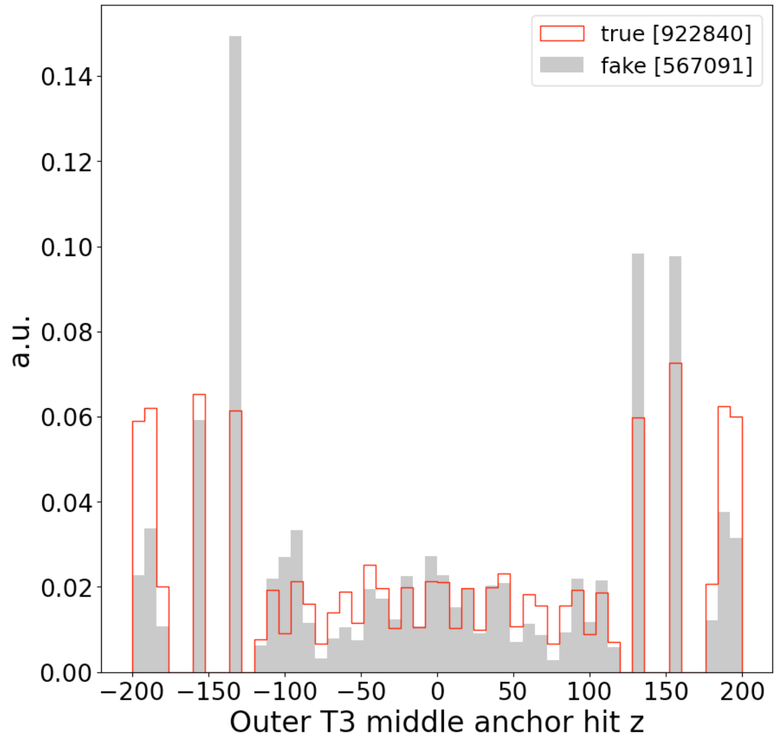


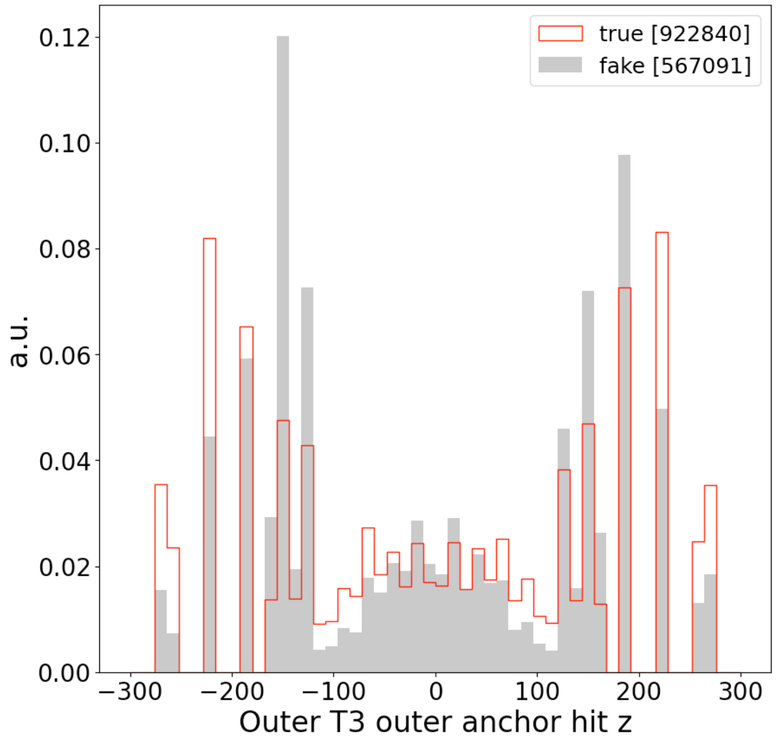
Trial 2: Input Features

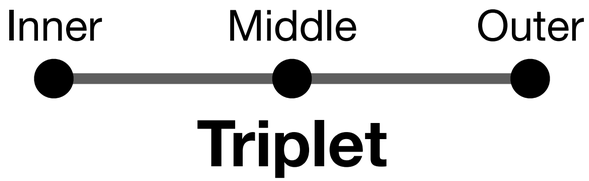
All plots available here





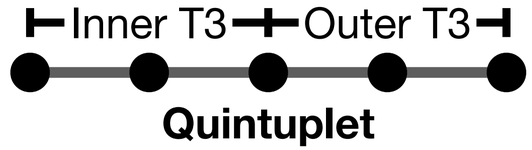


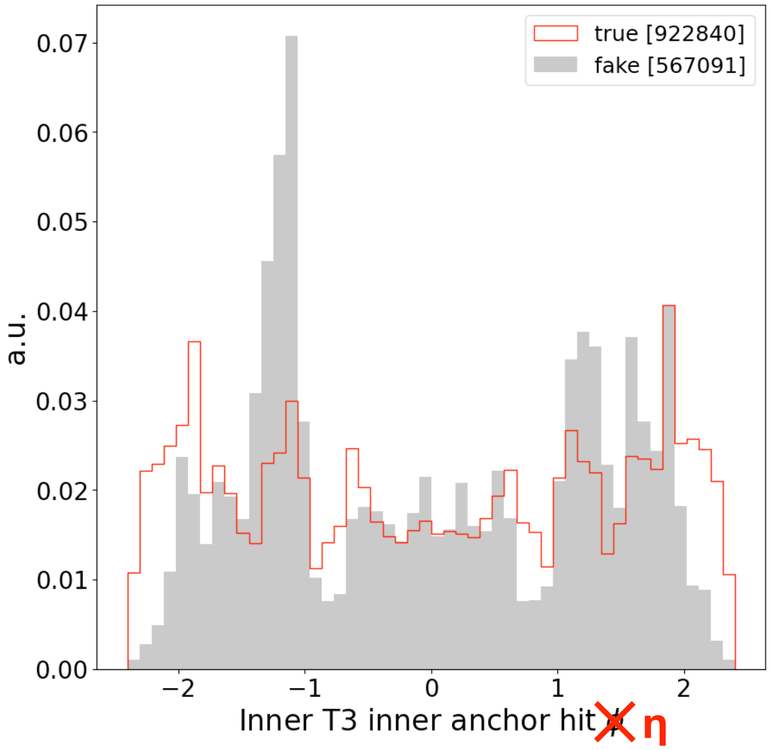


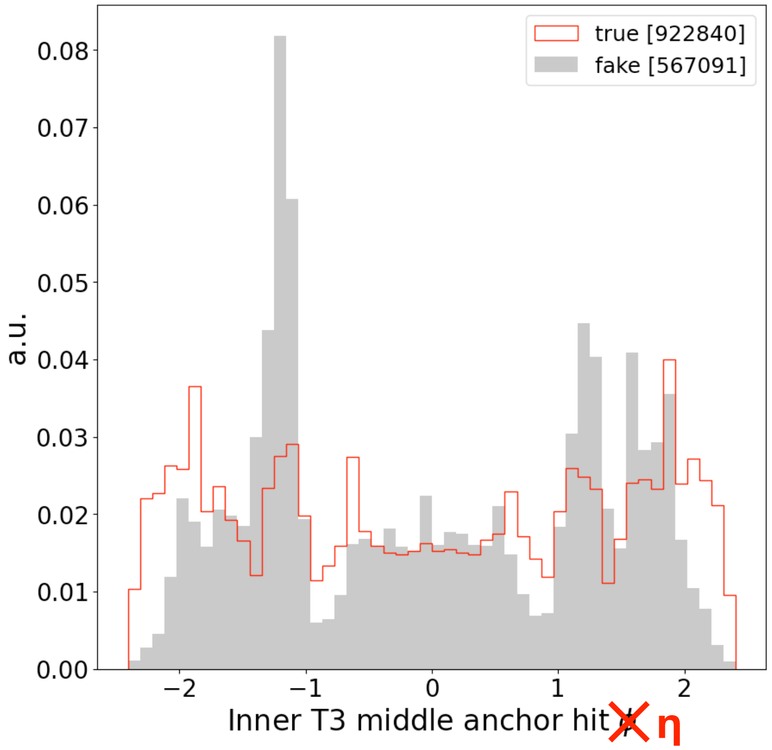


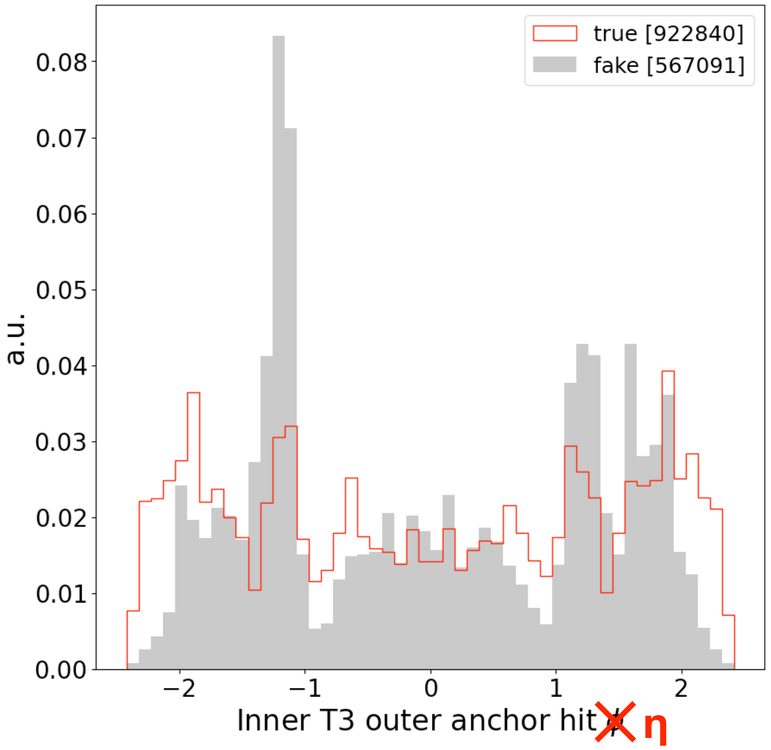
Trial 2: Input Features

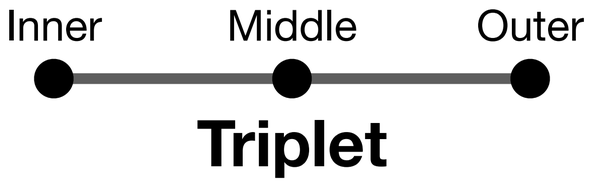
All plots available here





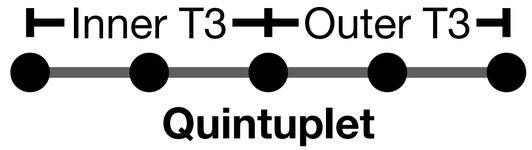


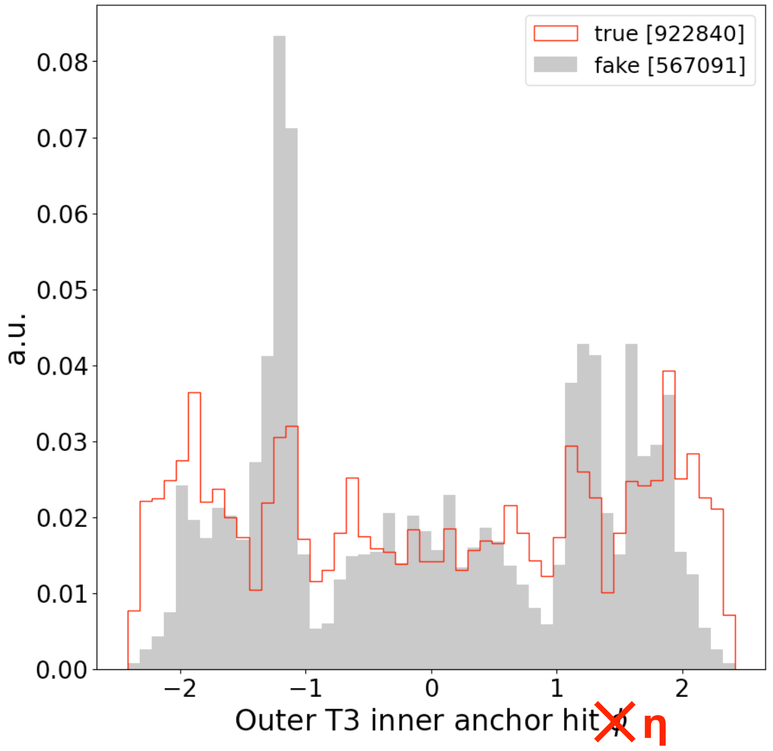


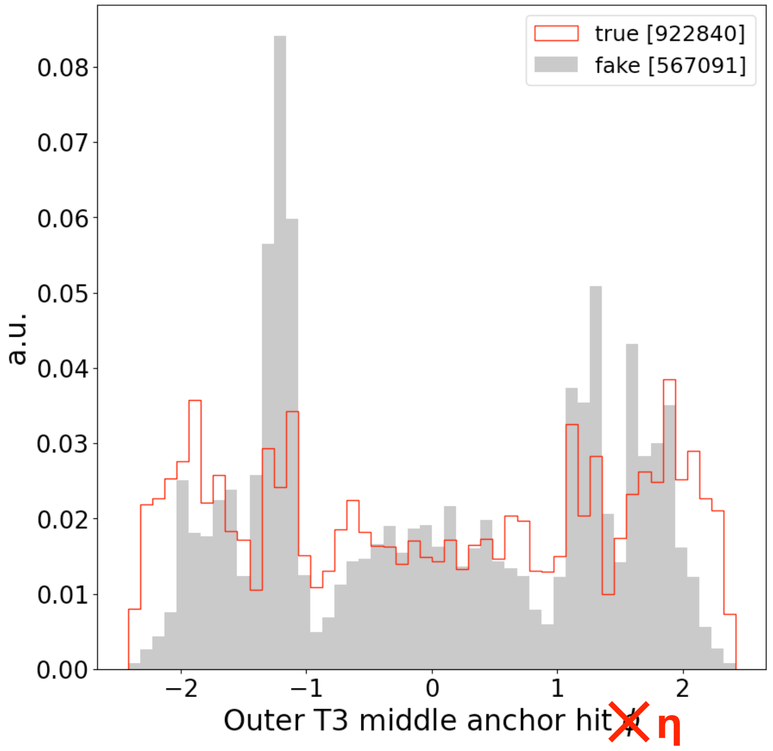


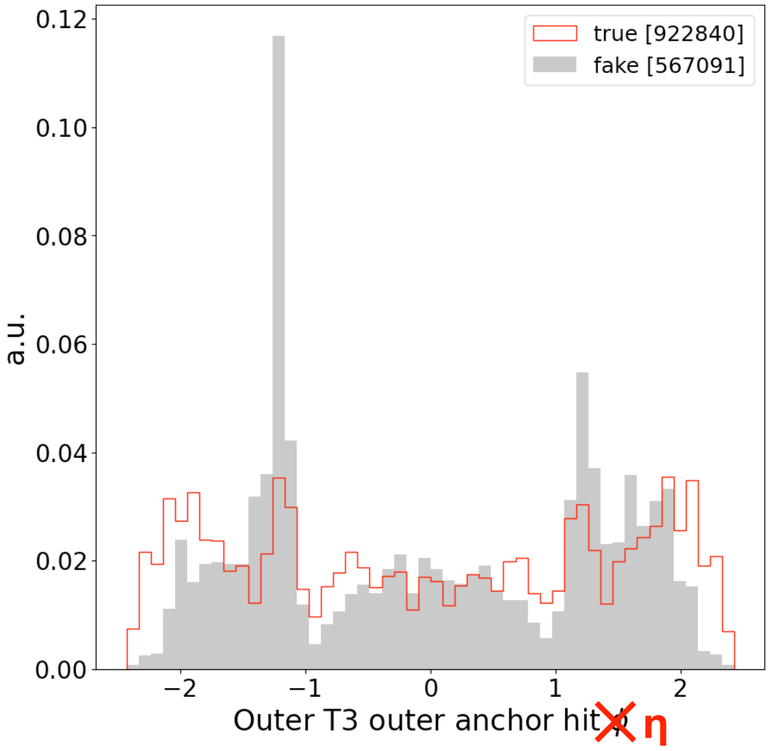
Trial 2: Input Features

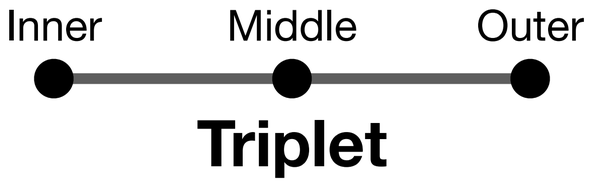
All plots available here





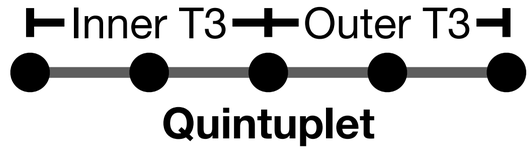


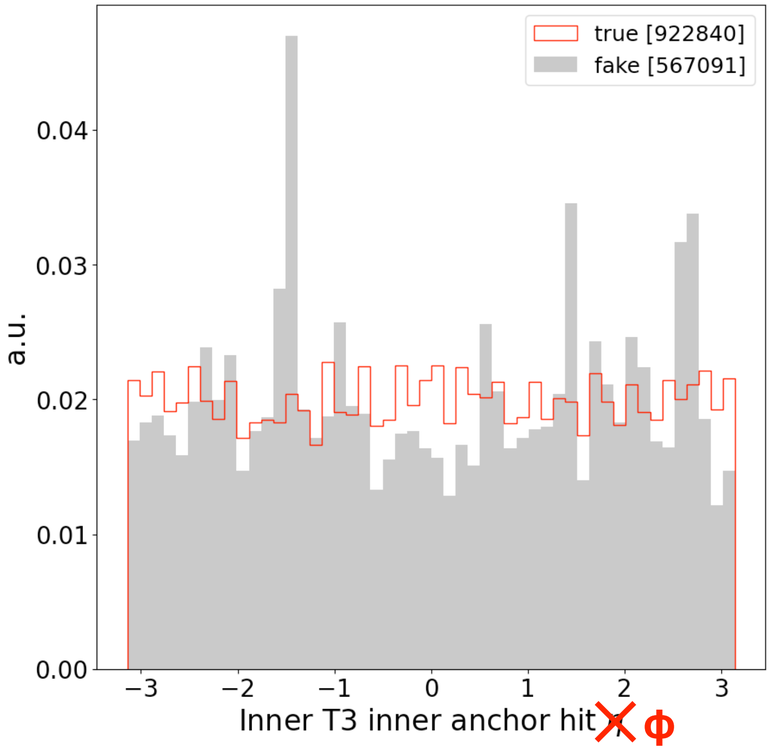


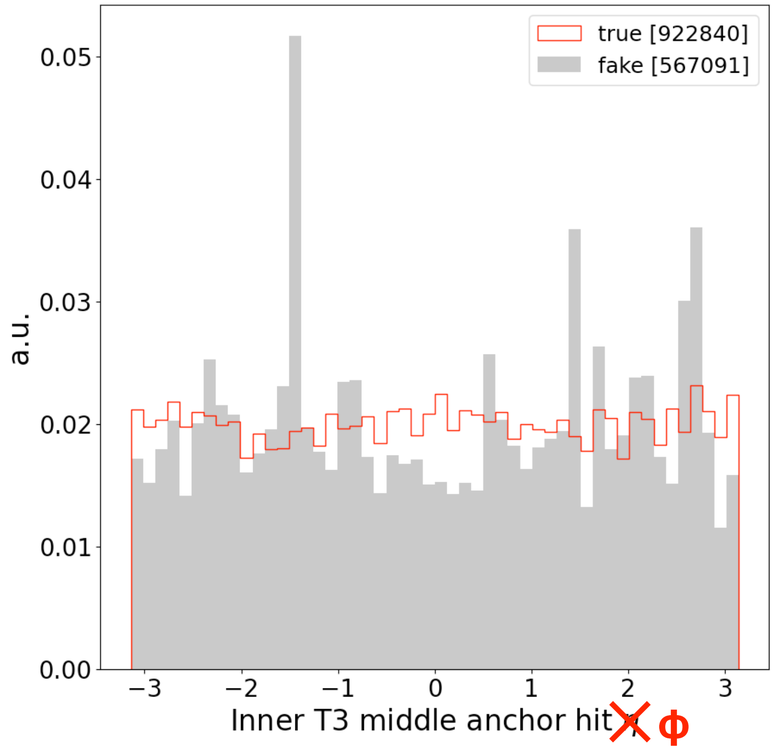


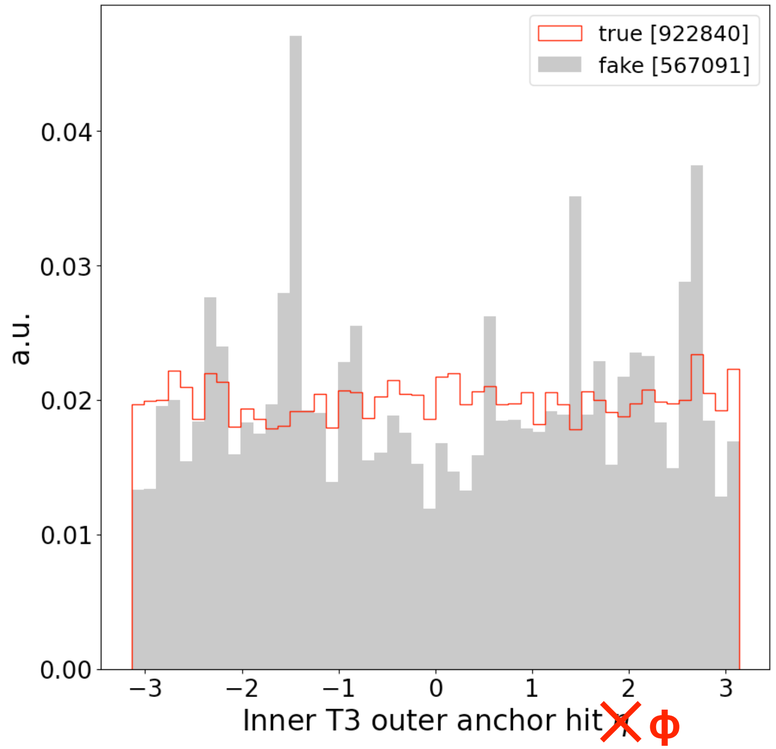
Trial 2: Input Features

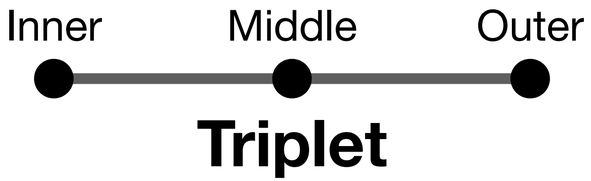
All plots available here





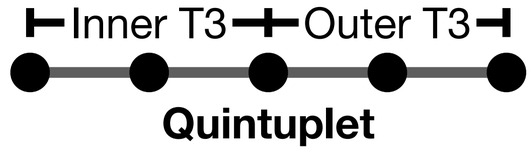


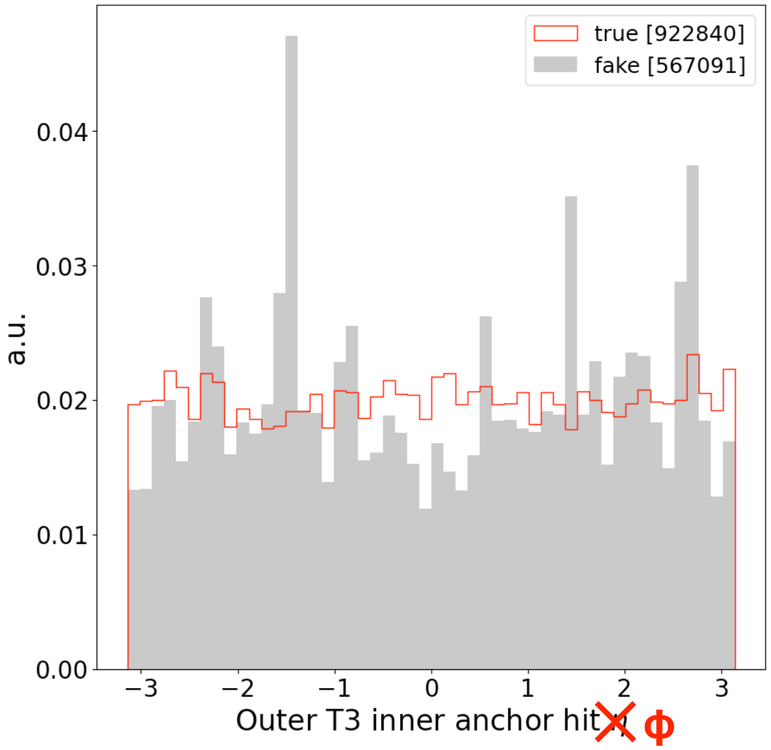


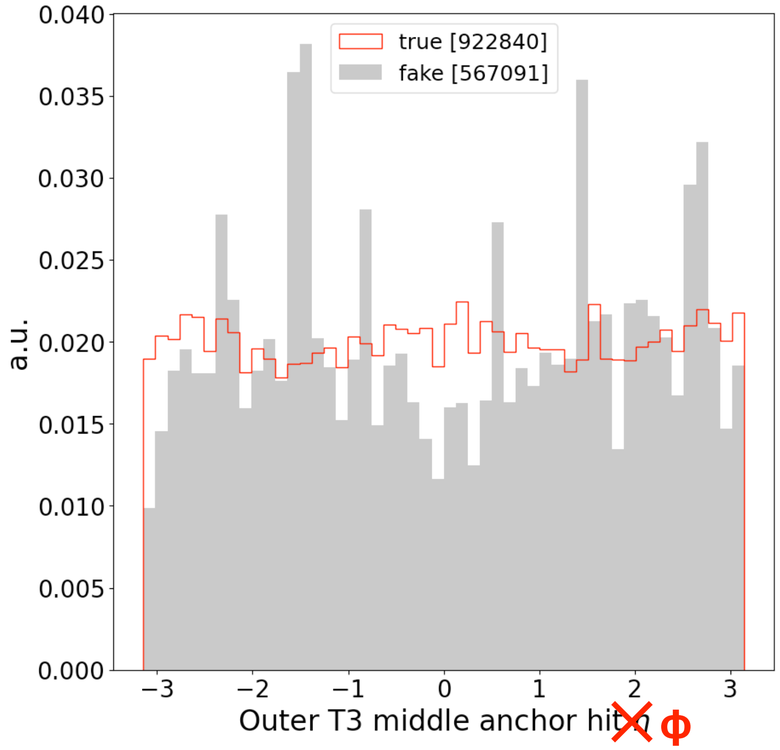


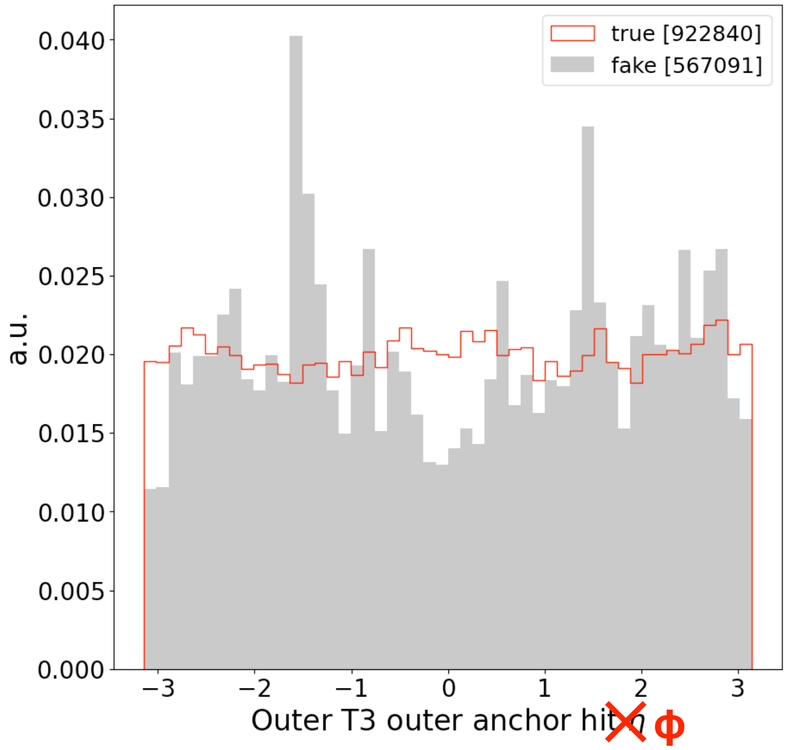
Trial 2: Input Features

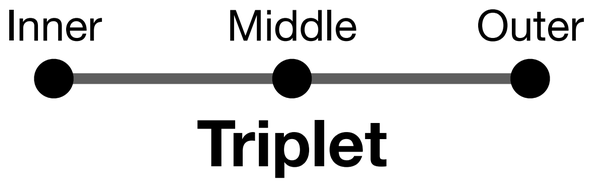
All plots available here





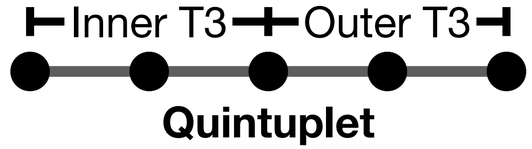


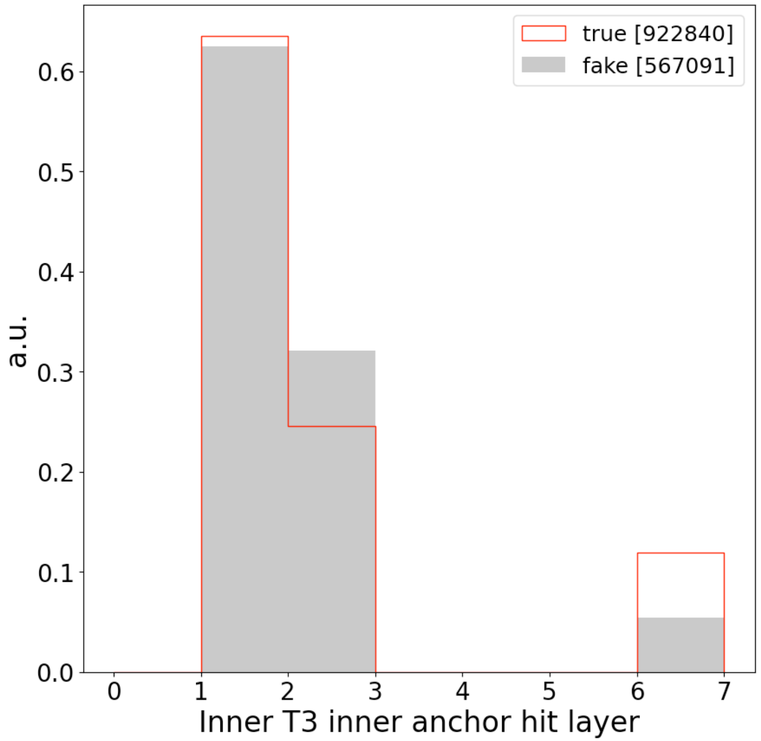


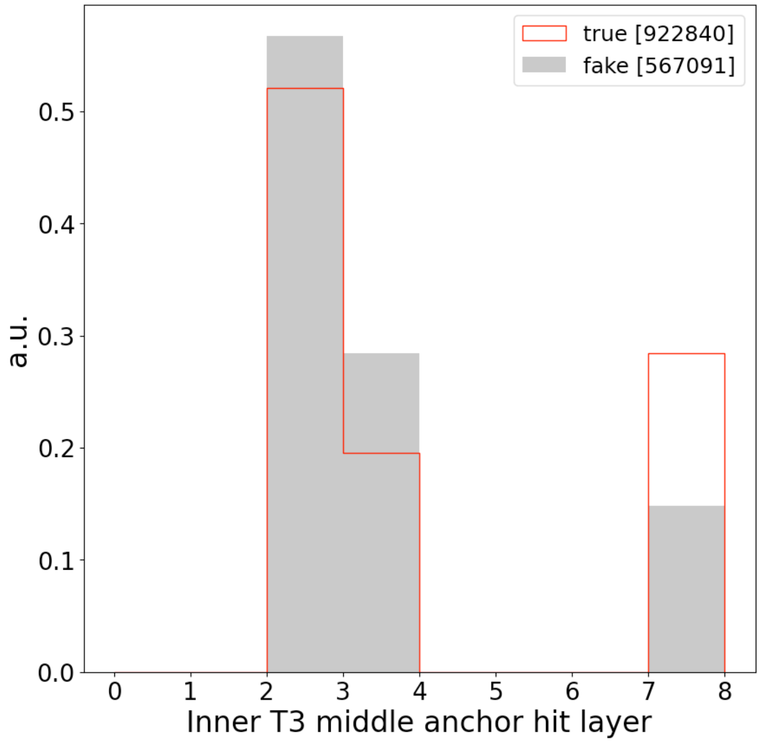


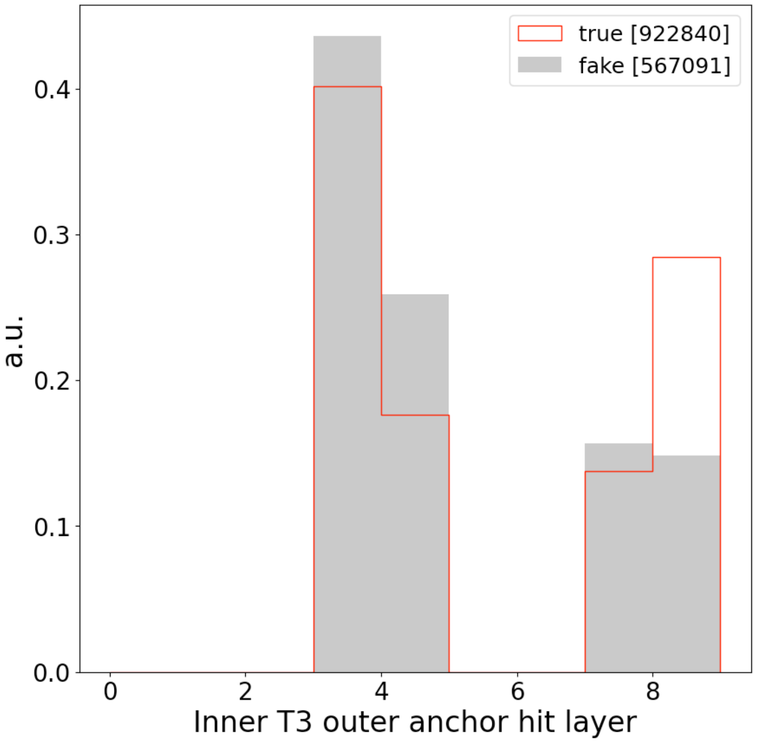
Trial 2: Input Features

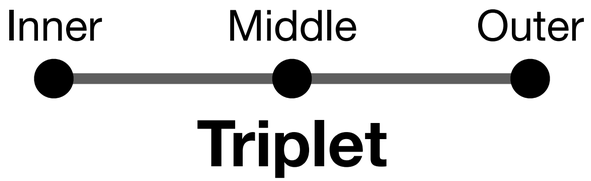
All plots available here





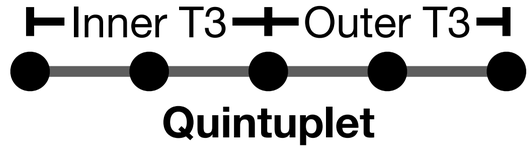


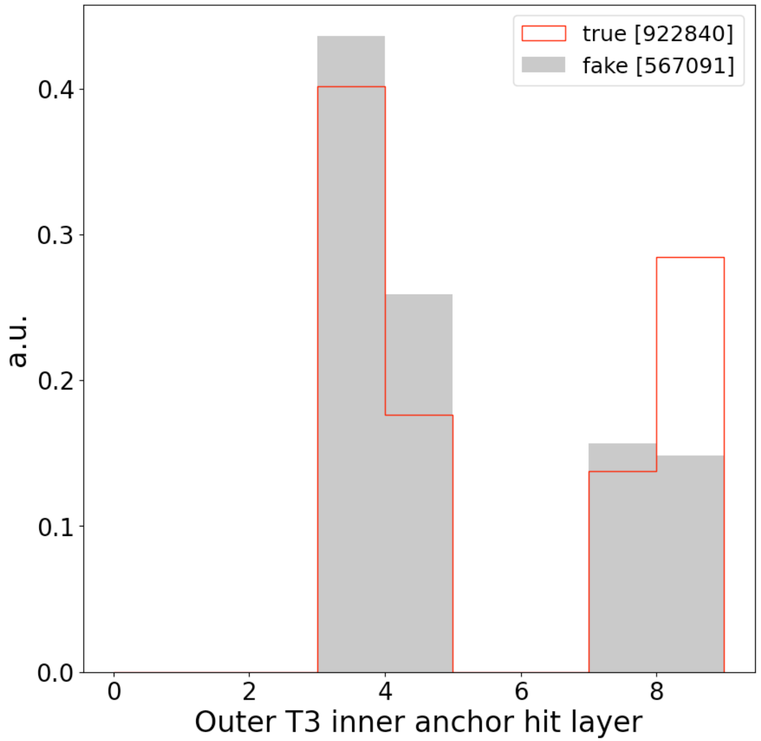


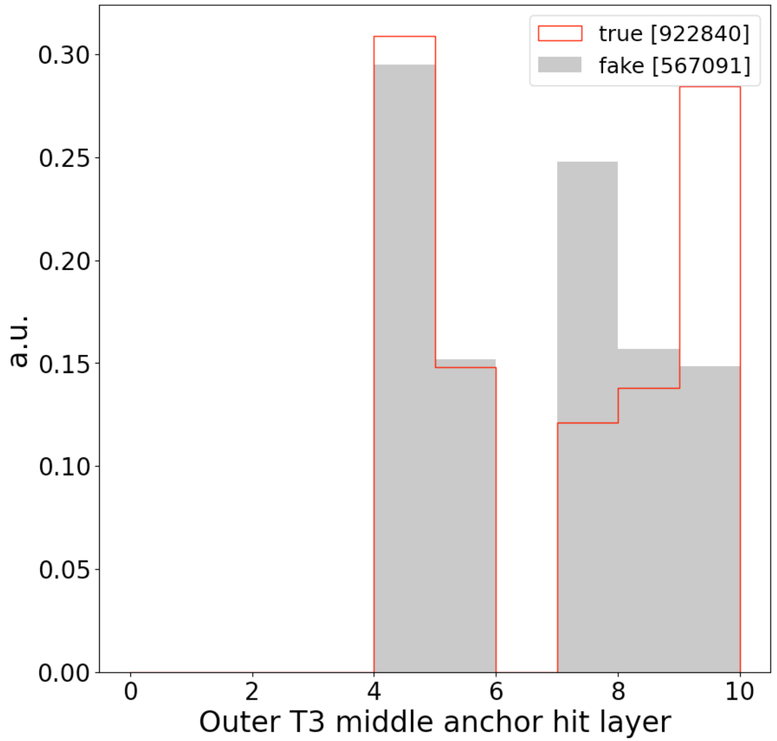


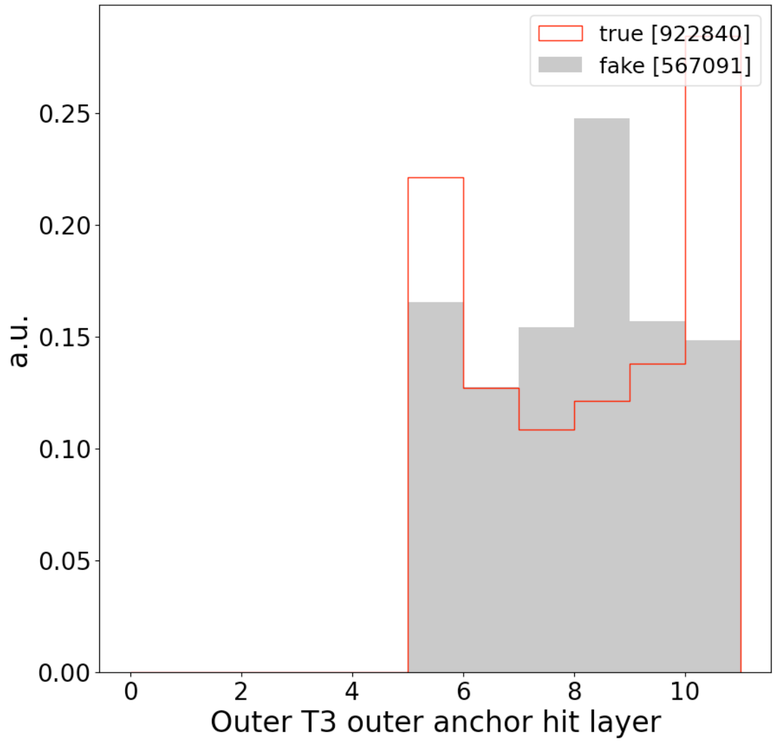
Trial 2: Input Features

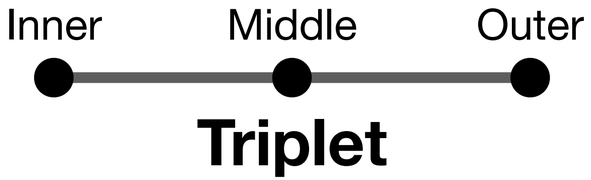
All plots available here







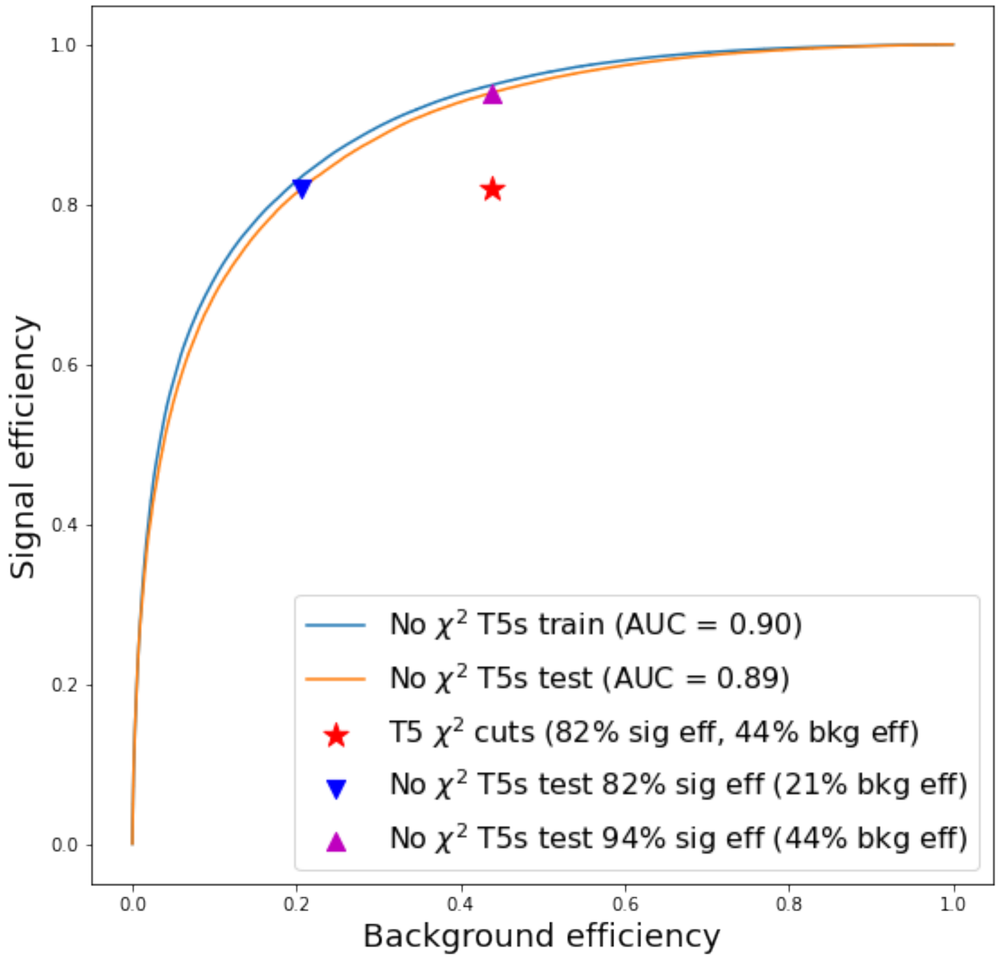




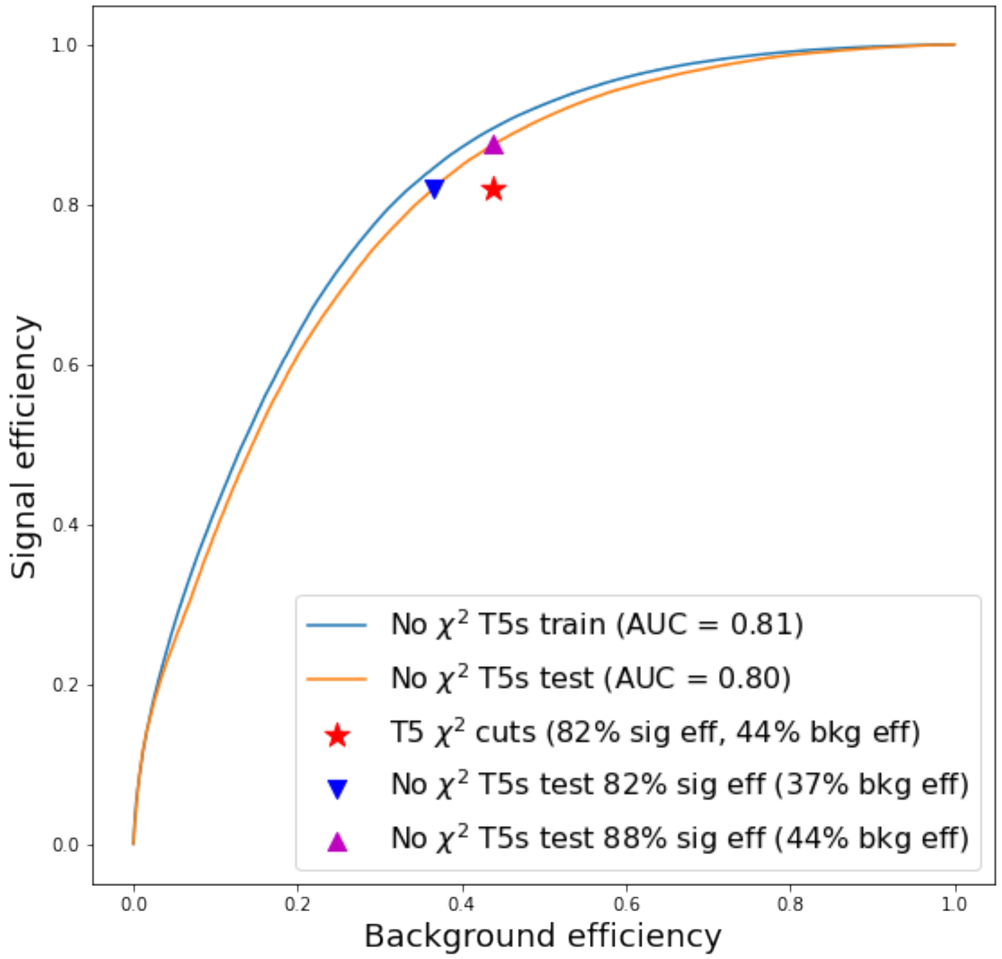
Trial 2: ~~GNN~~DNN Performance

1-layer DNN outperforms NN (NN ≈ χ^2 cuts)

1 hidden layer



No hidden layers



Summary

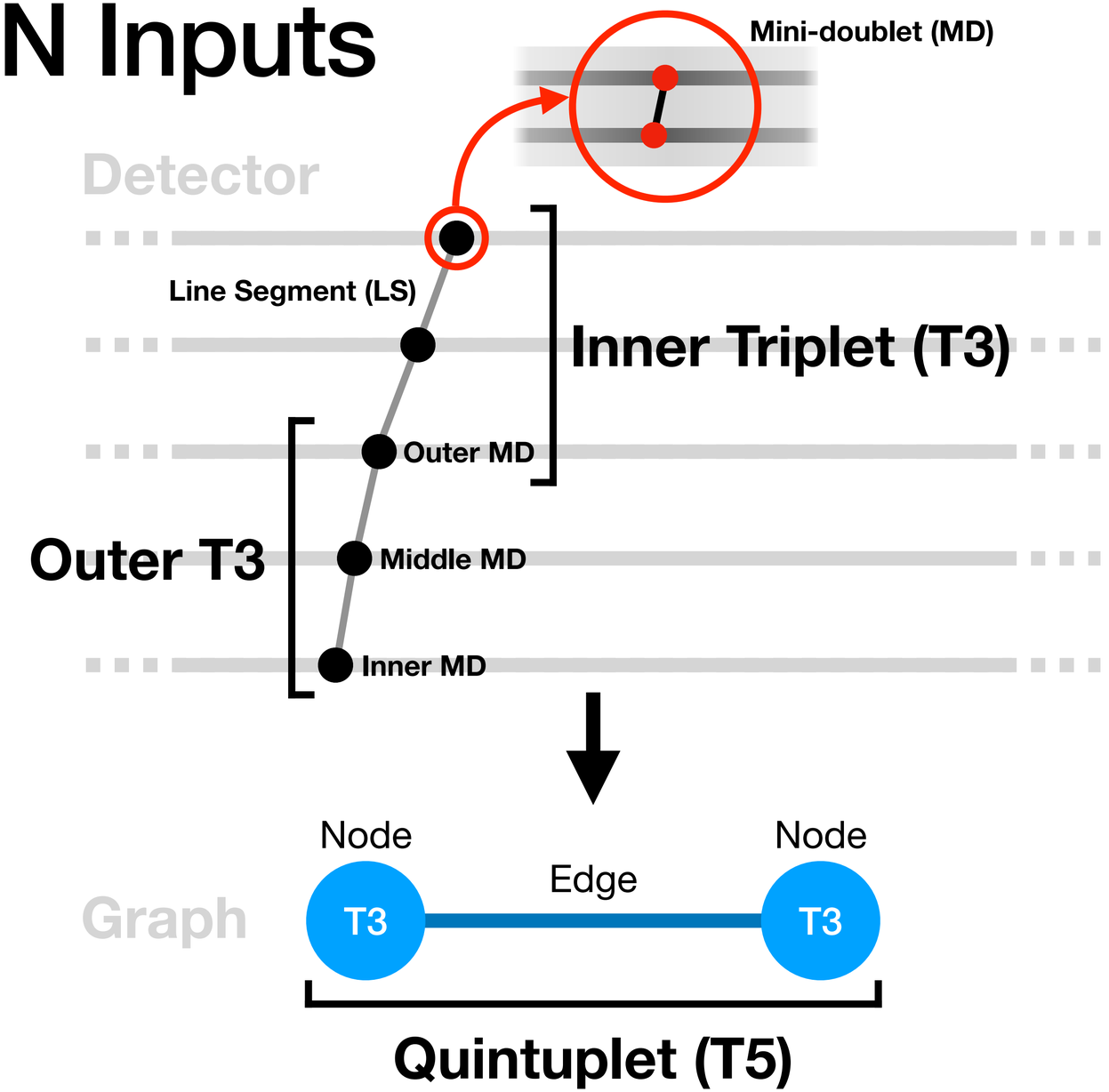
- For Trial 1, the DNN (1 hidden layer) and NN (no hidden layers) perform equivalently
- For Trial 2, the DNN beats the existing χ^2 cuts while the NN does not help as much
  - Still need to see how this impacts the final T5s in the TC collection
- Next steps:
  - Port the Trial 1 DNN/NN scores over to LST and create efficiency plots
  - Do a preliminary port of the Trial 2 DNN/NN to LST:
    - Should be able to write the matrix multiplication out “by hand”

Backup

GNN/DNN Inputs

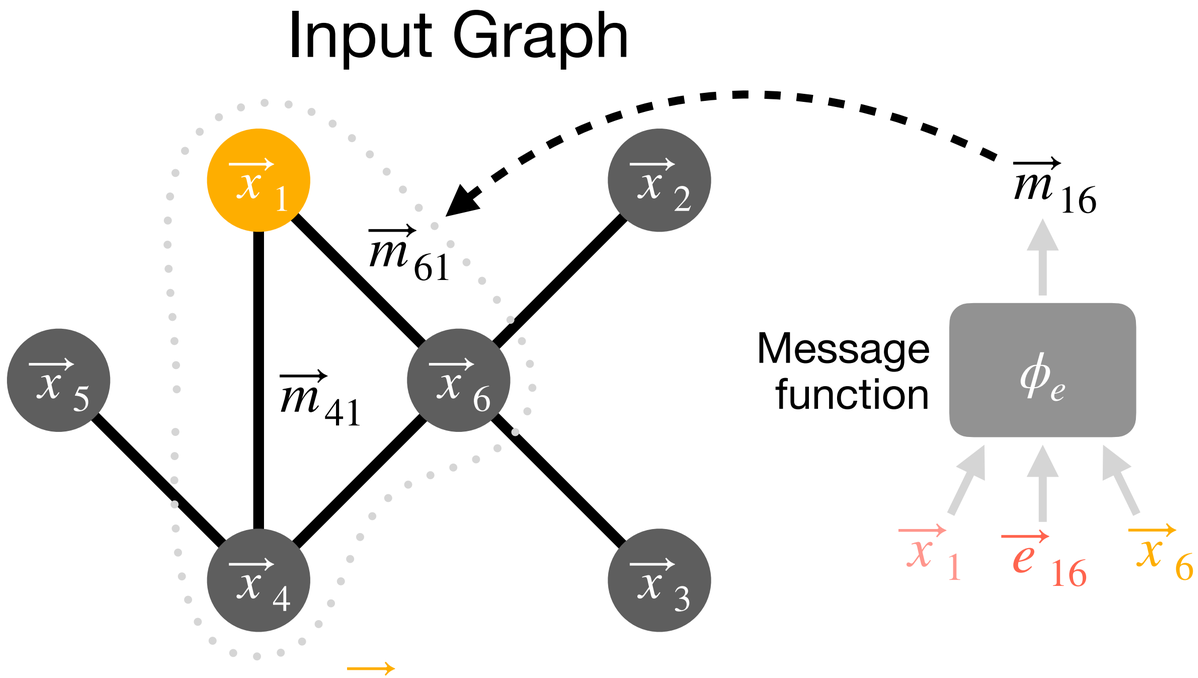
Object	Feature
Node (T3)	p_T
	Inner anchor hit r, z, ϕ, η, δr, layer
	Middle anchor hit r, z, ϕ, η, δr, layer
	Outer anchor hit r, z, ϕ, η, δr, layer
Edge (T5)	p_T, η, ϕ
	χ^2
	Inner T3 radius
	Bridge T3 radius
	Outer T3 radius

Scaled such that all features ∈ [0, 1]

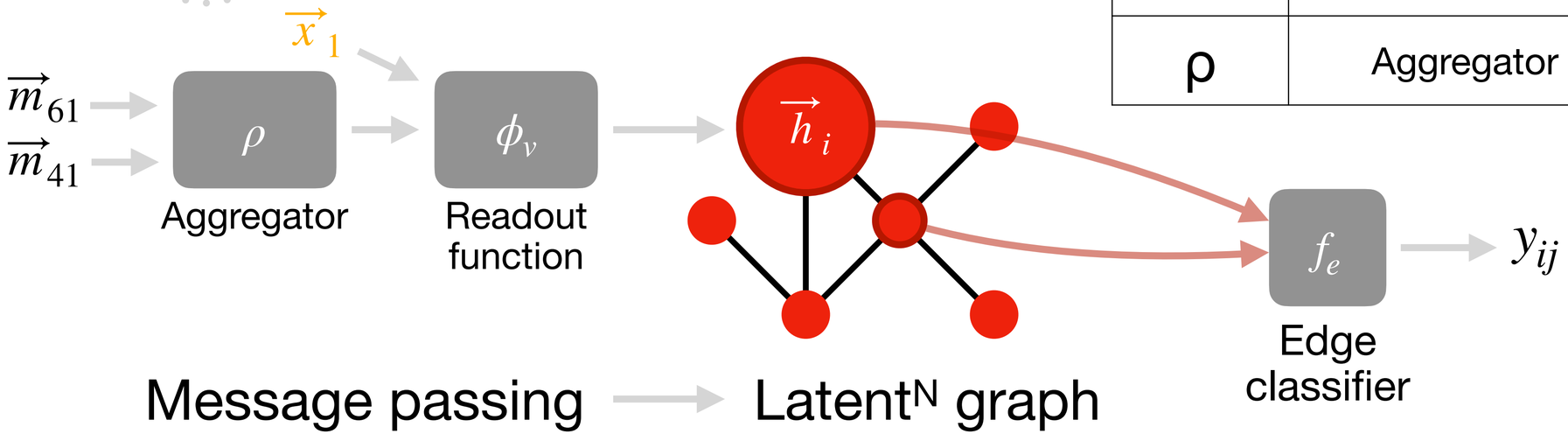


UC San Diego

GNN Configuration

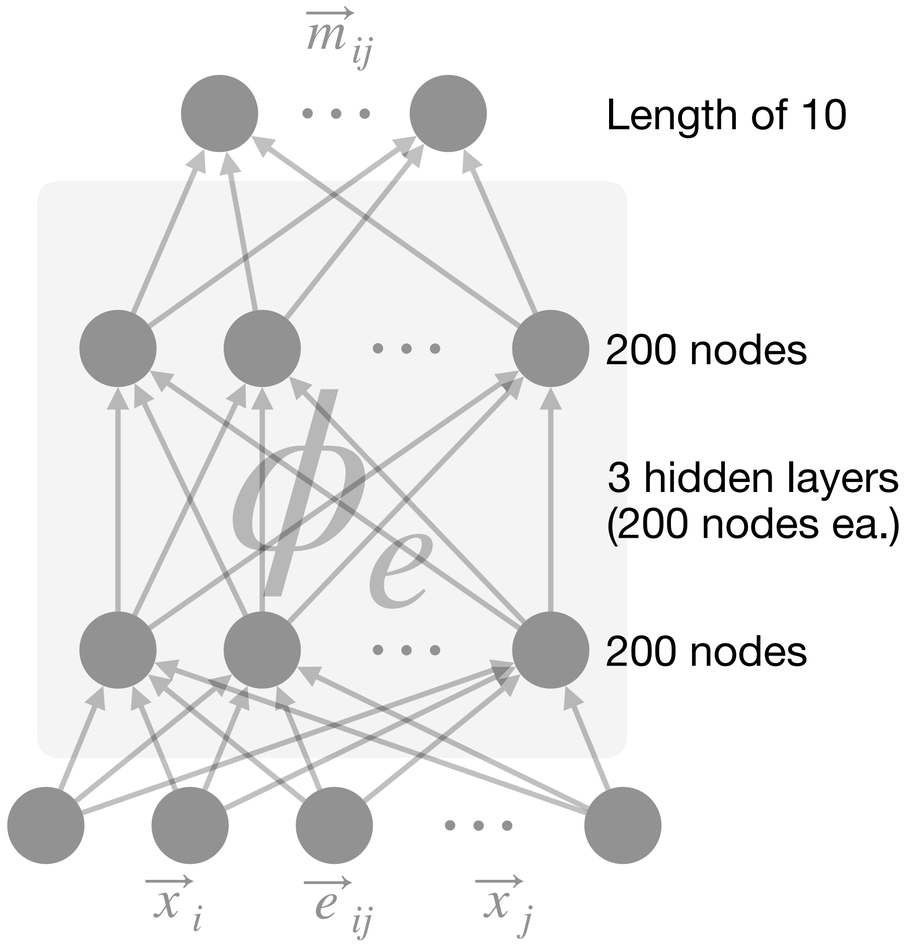


Symbol	Name	Description
ϕ_e	Message function	• Neural network • 4 hidden layers • 64 nodes per layer
ϕ_v	Readout function	• Neural network • 4 hidden layers • 64 nodes per layer
f_e	Edge classifier	• Neural network • 4 hidden layers • 64 nodes per layer
ρ	Aggregator	Sum

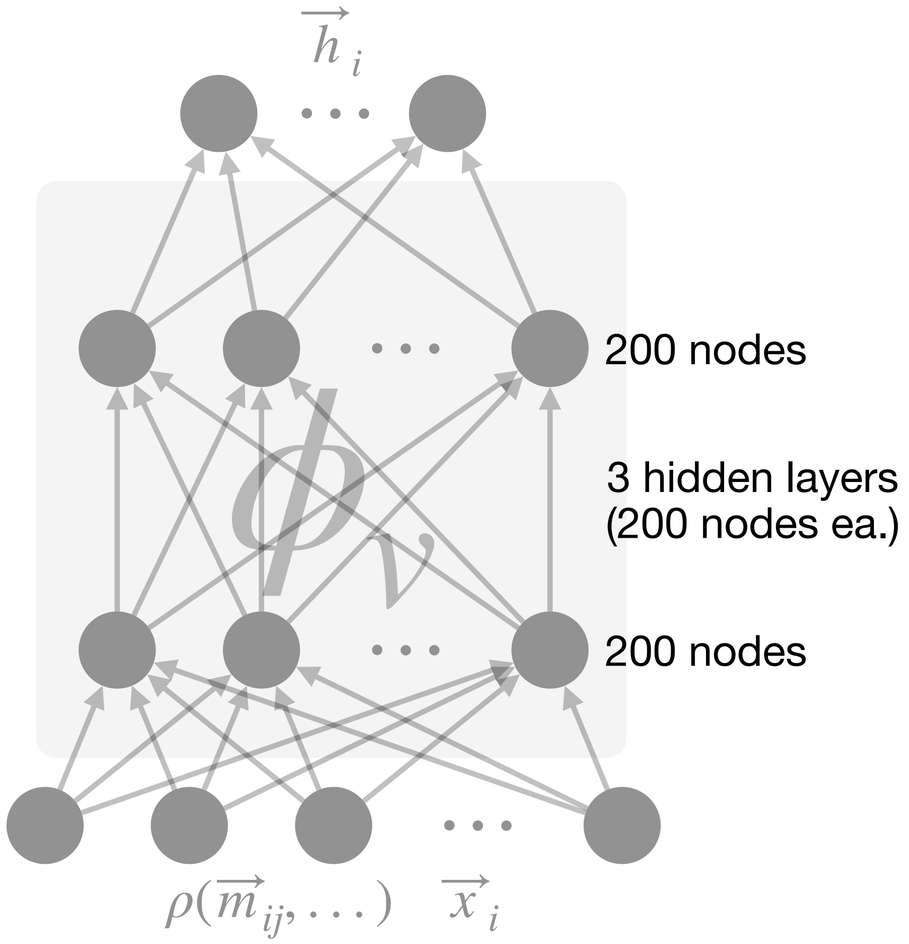


UC San Diego

GNN Internal NN Configurations



Message function



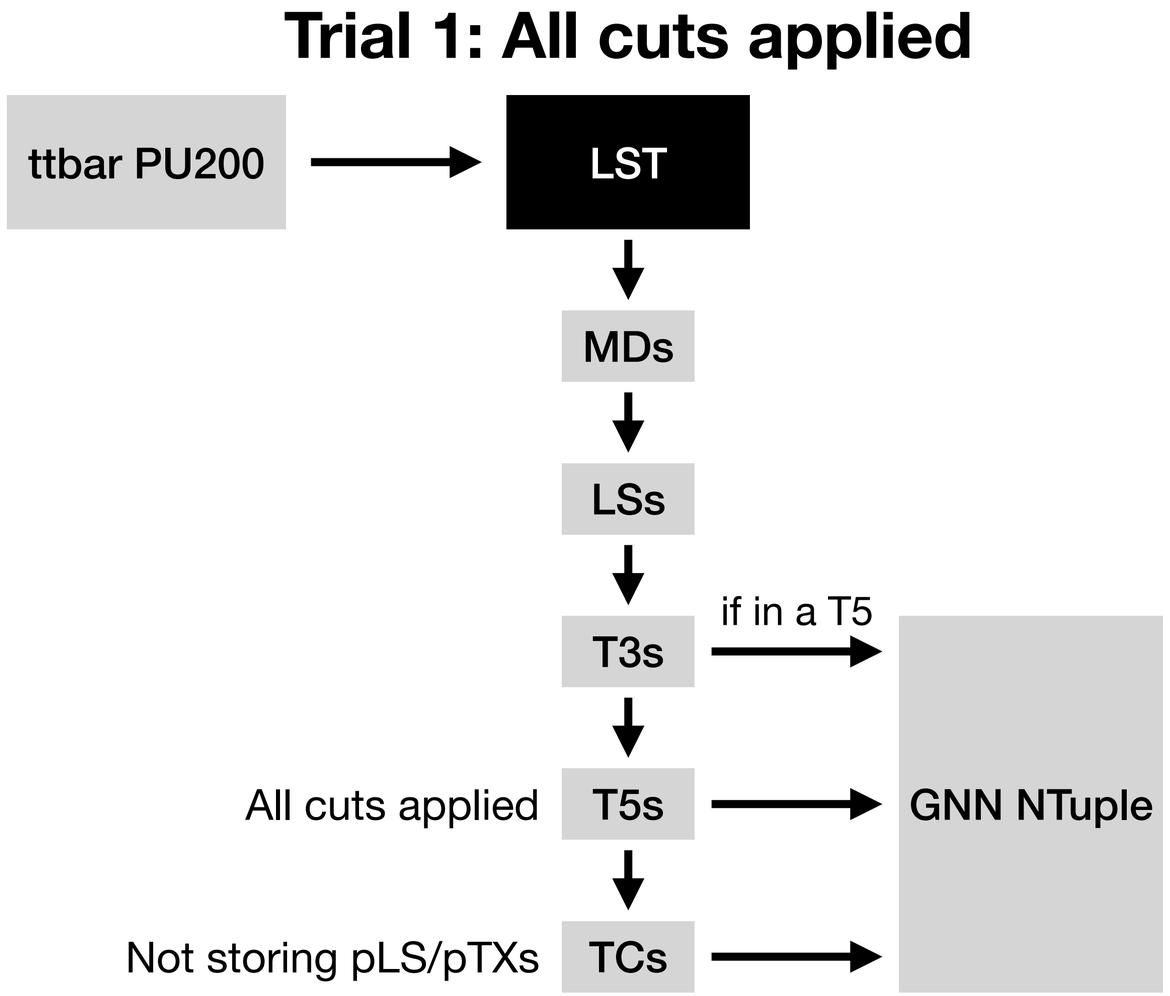
Readout function

UC San Diego

Trial 1: GNN NTuple

- Training on T5s/T3s without duplicate removal
- pLS/pTX objects are excluded from the TCs
  - i.e. only T5s in TCs
  - Duplicate removal is applied here
- **All cuts applied in T5 selection algo.**
- Target question:

Can the GNN give us anything for free?



UC San Diego