Enforcing Self-Consistent Kinematic Constraints in Neutrino Energy Estimators

Using GENIE Atmospheric Events

NuFact 2024

by J. L. Barrow
The University of Minnesota
with a Special Thanks to
S. Wu, R. R. Richi, T. Thakore, C. Borden, & M. Rabelhofer

September 18th, 2024

Chief Collaborators

Shaowei Wu
Uni. of Minnesota
Graduate Student
Network Architecture and transformer code development, ML expertise

Miranda Rabelhofer
Formerly IU
Postdoc
DUNE/NOvA data prep., ML expertise

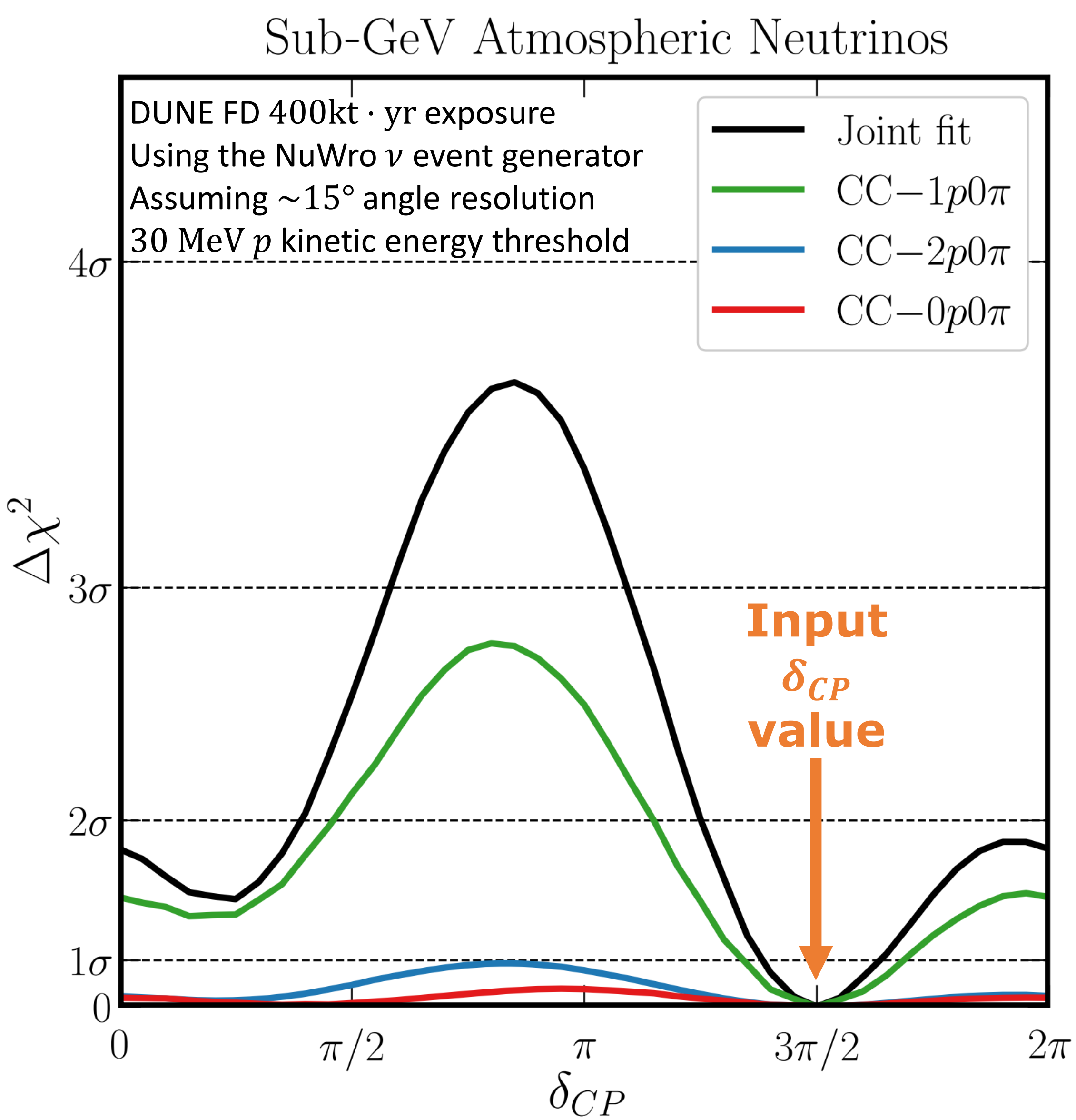
Raisa Rahman Richi
Franklin & Marshall College
FNAL SIST UG Intern
GENIE simulated data preparation, EAF job management, plotting

Tarak Thakore
Formerly U. Cinci.
URA Visiting Scholar Postdoc
DUNE data prep., Conda environment, burgeoning ML expertise

Casey Borden
IU
Incom. Grad. Stud.
Native ROOT to Pandas dataframe conversion, plotting

Thanks to many, many more for discussions!
Care to join us?

Some Recent Realizations…



Atmospheric ν oscillation sensitivities got me thinking…

- Machado, Kelly, Martinze-Soler *et al*: [Phys. Rev. Lett. 123, 081801 (2019)]

- A key point at low energies…
  - **Angle reco. is sensitivity driver**
  - Need to point well to get L
    - E doesn't matter as much
  - $\sim 3\sigma$ δ_{CP} sensitivity w/~15° pointing…
  - $\sim 5\sigma$ δ_{CP} sensitivity with ~7° pointing…?

- Use primarily $\mathrm{CC}1\mu 1\mathrm{p}0\pi$ interactions

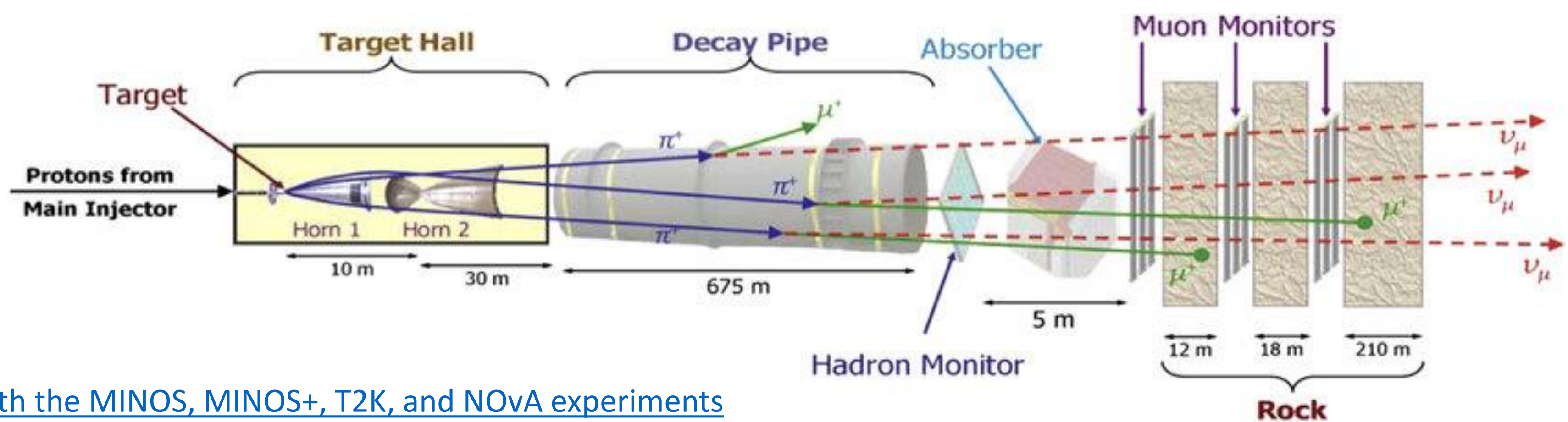
- **Atmospherics to be DUNE's first physics measurement**
  - Beam won't be active for ~2 years…

- **Why am I telling you all this?**

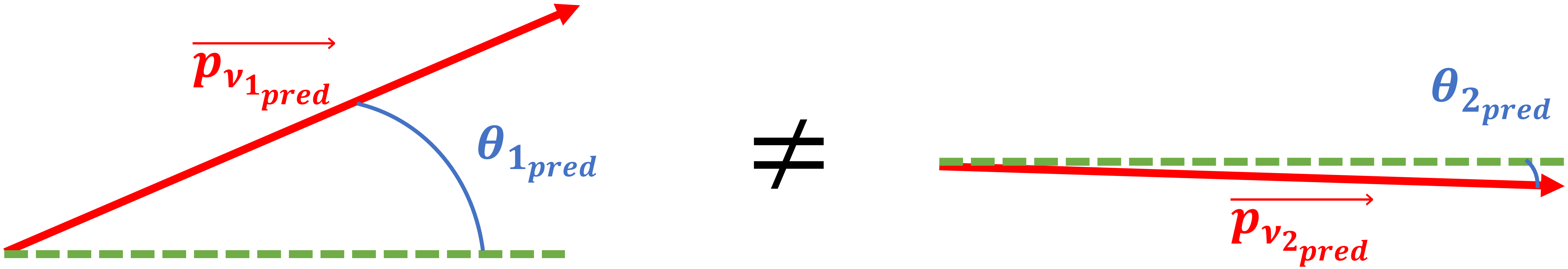
Sub-GeV Atmospheric Neutrinos and CP Violation in DUNE

How to Potentially Improve Energy Estimators?

- **For accelerator-based energy estimators:**
  - Many E_ν estimators are likely kinematically incomplete…
  - The key thing is *the momentum as a vector* $\overrightarrow{p_\nu}$, not only $|\overrightarrow{p_\nu}|$
    - Implies need to reconstruct both *magnitude* and *direction* well
  - **~Ignoring kinematic/angular constraint on E_ν**
    - Given known angular resolution & some ~nuclear physics smearing…
    - …and some true incoming angular spread (divergence)…
    - …should know when to *reject* an energy prediction from estimator
- If energy estimator is incorrect, we continue training…
  - If energy estimator *seems* correct…
    - Check if consistent w/angle! **Minimize loss with energy <u>&</u> angle!**



[Neutrino oscillations with the MINOS, MINOS+, T2K, and NOvA experiments]



$$\boldsymbol{E_{\nu_{1_{pred}}} = E_{\nu_{2_{pred}}}}$$

$$\overrightarrow{p_{\nu_{1_{pred}}}} \neq \overrightarrow{p_{\nu_{2_{pred}}}}$$

$$\theta_{1_{pred}} \neq \theta_{2_{pred}}$$

We should continue training until things are kinematically consistent!

Must include <u>new inputs to loss function</u>!

"Consider a neutrino of initial true energy E_ν whose incoming angle lies along the green dotted line. Consider a ML algorithm's output derived from two independent (stochastically differentiable) trainings, each utilizing a simplified loss function of the form $L\left(E_\nu^{true}, E_\nu^{pred}\right)$. The predicted energy of an incoming neutrino of true momentum p_ν comes from the magnitude $E_\nu \equiv |p_\nu|$. If training 1 and training 2 end after achieving similar loss or accuracy criteria (which are dependent only on variables of energy), even if each predicts very similar scalar energies $E_\nu \approx \left|p_{\nu_{1_{pred}}}\right| \approx \left|p_{\nu_{2_{pred}}}\right|$ for an incoming neutrino on an event-by-event basis, there is no automatic guarantee that angular correlations are respected in a kinematically consistent way $\theta_\nu \napprox \theta_{1_{pred}} \napprox \theta_{1_{pred}}$ without a loss function which penalizes such behavior."

Argument as follows…

- Many (not all) ML kinematic estimators optimize a single loss function of a single variable, s_ν:

$$L\left(s_\nu^{true}, s_\nu^{pred}\right) = \cdots$$

  - L here can be any particular style of loss function…
    - Mean square error, mean absolute error, mean absolute % error…

- We seek to encourage learning by instead making the loss composite and multivariate on possibly many kinematic variables simultaneously, $s_\nu,\ x_\nu, \ldots$

$$\mathcal{L}\left(\left\{s_\nu^{true}, s_\nu^{pred}\right\}, \left\{x_\nu^{true}, x_\nu^{pred}\right\}, \ldots\right) = \alpha L_s\left(s_\nu^{true}, s_\nu^{pred}\right) + \beta L_x\left(x_\nu^{true}, x_\nu^{pred}\right) + \cdots$$

  - Can use this to imbue physics into the loss function
    - "Physics motivated loss functions"
    - "Physics informed machine learning" (PIML)

Argument as follows…

- Many (not all) ML kinematic estimators optimize a single loss function of a single variable, s_ν:

$$L\left(s_\nu^{true}, s_\nu^{pred}\right) = \cdots$$

  - L here can be any particular style of loss function…
    - Mean square error, mean absolute error, mean absolute % error…

- We seek to encourage learning by instead making the loss composite and multivariate on possibly many kinematic variables simultaneously, s_ν, x_ν, …

$$\mathcal{L}\left(\left\{s_\nu^{true}, s_\nu^{pred}\right\}, \left\{x_\nu^{true}, x_\nu^{pred}\right\}, \ldots\right) = \alpha L_s\left(s_\nu^{true}, s_\nu^{pred}\right) + \beta L_x\left(x_\nu^{true}, x_\nu^{pred}\right) + \cdots$$

$$\rightarrow \mathcal{L}\left(\left\{s_\nu^{true}, s_\nu^{pred}\right\}, \left\{x_\nu^{true}, x_\nu^{pred}\right\}, \ldots\right) \rightarrow \mathcal{L}\left(\left\{E_\nu^{true}, E_\nu^{pred}\right\}, \bigcup_{i=1}^{3}\left\{p_{\nu_i}^{true}, p_{\nu_i}^{pred}\right\}, \ldots\right)$$

$$\rightarrow \mathcal{L}\left(\left\{E_\nu^{true}, E_\nu^{pred}\right\}, \bigcup_{i=1}^{3}\left\{p_{\nu_i}^{true}, p_{\nu_i}^{pred}\right\}, \ldots\right) = \alpha L_E\left(E_\nu^{true}, E_\nu^{pred}\right) + \sum_{i=1}^{3}\beta_i L_i\left(p_{\nu_i}^{true}, p_{\nu_i}^{pred}\right) + L(\cdots) + \cdots$$

Past "Composite" Losses w/LSTM

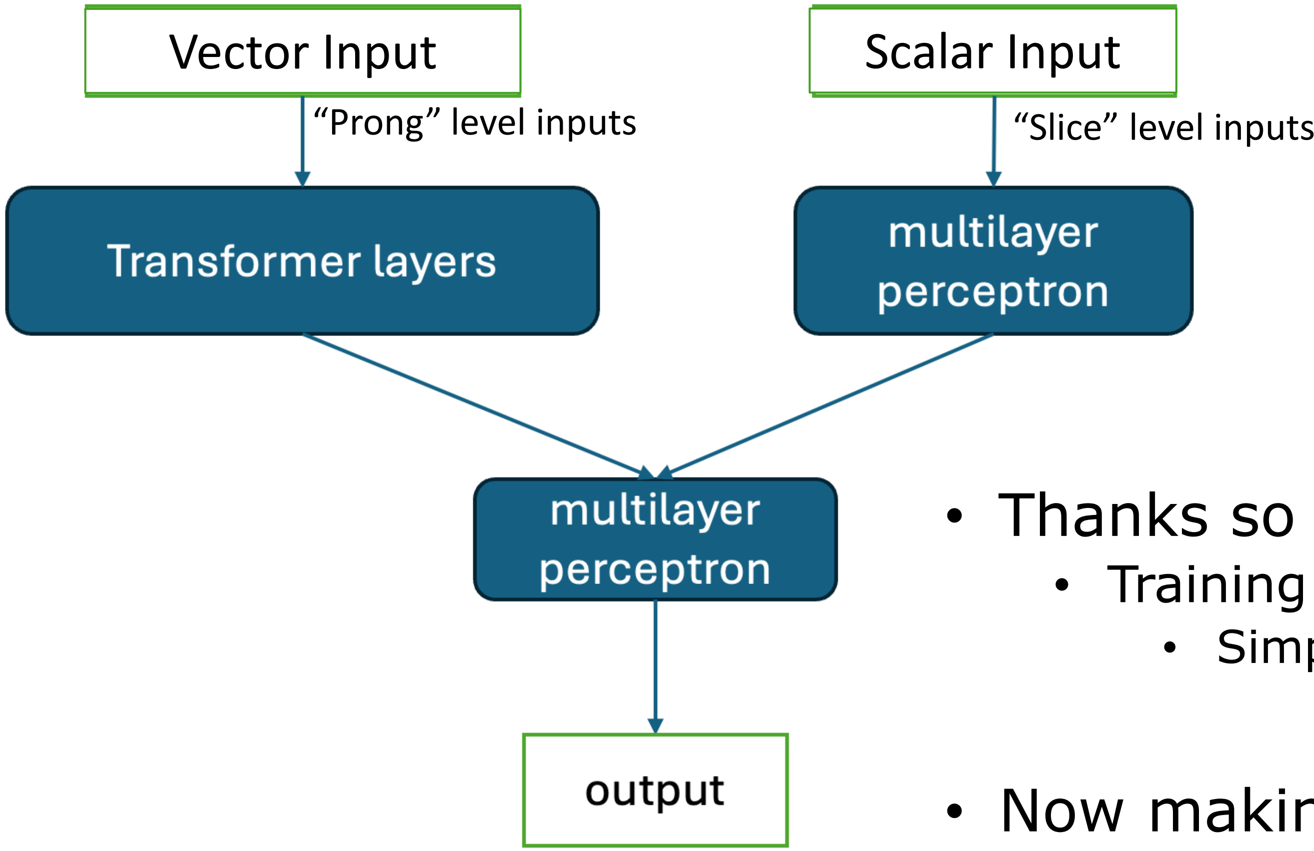
- Long Short-Term Memory (LSTM) NN architecture
  - Previously utilized for ν and ℓ energy estimation in NOvA
    - A. Sutton, Domain Generalization with Machine Learning in the NOvA Experiment
    - D. Torbunov, Improving Energy Estimation at NOvA with Recurrent Neural Networks
    - D. Torbunov, BNL Seminar
  - Used something akin to…

$$\frac{1}{2}\left[L_E\left(E_\nu^{true}, E_\nu^{pred}\right) + L_\ell\left(E_\ell^{true}, E_\ell^{pred}\right)\right]$$

    - Permits basic understanding of E_{had} via *assumed* energy conservation (*not in loss*)
    - The LSTM was optimized for these two variables, down to its two-headed architecture
    - Can in principle go beyond two variables, but requires many changes (hard coded)
- LSTM development in NOvA taken over by Shaowei Wu
  - Initial application of LSTM to DUNE beam events followed
  - Initial developments made on a new model w/a transformer
    - This model architecture is more flexible, easily handling many loss variables at once
    - Shaowei redeveloped this model for our purposes, with eyes toward DUNE & NOvA

S. Wu

Transformer Network is Up & Running!



- Thanks so much to Shaowei!
  - Training has been successful
    - Simple stuff thus far
- Now making lots of data sets
  - Atmospherics first, beams soon

[wswxyq/transformer_EE github]

[tarak-thakore/transformer_EE github]

[tarak-thakore/transformer_EE at josh_develop github]

From Shaowei Wu

Data Preparation w/GENIE & DUNE

- Current data preparation requirements:
    - Convert some ROOT file to CSV
    - CSV columnated data is easily importable to Pandas
        - Columns represent particular branch variables
            - Scalar/Slice or Vector/Prong type inputs are available
                - Scalar example: Total visible energy
                - Vector example: All indexed track lengths for a given event
            - Losses computed only against scalar variables
        - Rows represent whole events
        - New columns/variables can be added at will
    - Model configured to take any number of columns as input
        - Used for training features or losses

R. R. Richi
GENIE events with thresholds and topological selections

T. Thakore
DUNE Events
Coming Soon!

C. Borden
Direct (Up)ROOT to Pandas data conversion
Coming Soon!

- Richi:
    - CSVs with topological/process selections on truth-level GENIE events w/LArTPC-motivated kinetic energy thresholds
- Tarak: CSVs from DUNE events using AnaTrees
- Casey: Work directly from DUNE AnaTree/CAFs

Previously Chosen Training Variables

- Scalars (event level variables)
  - Final_State_Lepton_PDG, Final_State_Lepton_Mass
  - Final_State_Lepton_Energy ***THIS IS CHEATING FOR NC***
  - Final_State_Lepton_Momentum_X…Momentum_Y…Momentum_Z

- Vectors
  - Final_State_Particles_PDG, Final_State_Particles_Mass
  - Final_State_Particles_Energy
  - Final_State_Particles_Momentum_X…Momentum_Y…Momentum_Z

- Targets (for loss function)
  - Initial_State_Neutrino_Energy, Initial_State_Neutrino_Momentum_X…Momentum_Y…Momentum_Z

Currently Chosen Training Variables

These are about to change…??

- Scalars (event level variables)
  - ~~Final_State_Lepton_PDG, Final_State_Lepton_Mass~~
  - ~~Final_State_Lepton_Energy ***THIS IS CHEATING FOR NC****~~
  - ~~Final_State_Lepton_Momentum_X…Momentum_Y…Momentum_Z~~
  - KE_{Tot}, P_{Tot}, $P_{miss_{simple}}$ (*a la* A. Furmanski, UMN)

- Vectors—**NOW INCLUDING VISIBLE LEPTONS**
  - Final_State_Particles_PDG, Final_State_Particles_Mass
  - Final_State_Particles_Energy
  - Final_State_Particles_Momentum_X…Momentum_Y…Momentum_Z

- Targets (for loss function)
  - Initial_State_Neutrino_Energy, Initial_State_Neutrino_Momentum_X…Momentum_Y…Momentum_Z
  - Initial_State_Neutrino_Momentum_X…Momentum_Y…Momentum_Z

- Will show that this may have not been a great idea…
  - Future: Utilize CC & NC classifier, move back to separating lepton
    - Potentially directly classify within the new model with the transformer
  - Future: Go for both $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$ and $(E_\ell, p_{x_\ell}, p_{y_\ell}, p_{z_\ell})$ together *a la* LSTM loss?

GENIE-*Only* Results

Many Plots Shown Here from Richi

Using oscillated atmospheric neutrinos at Homestake

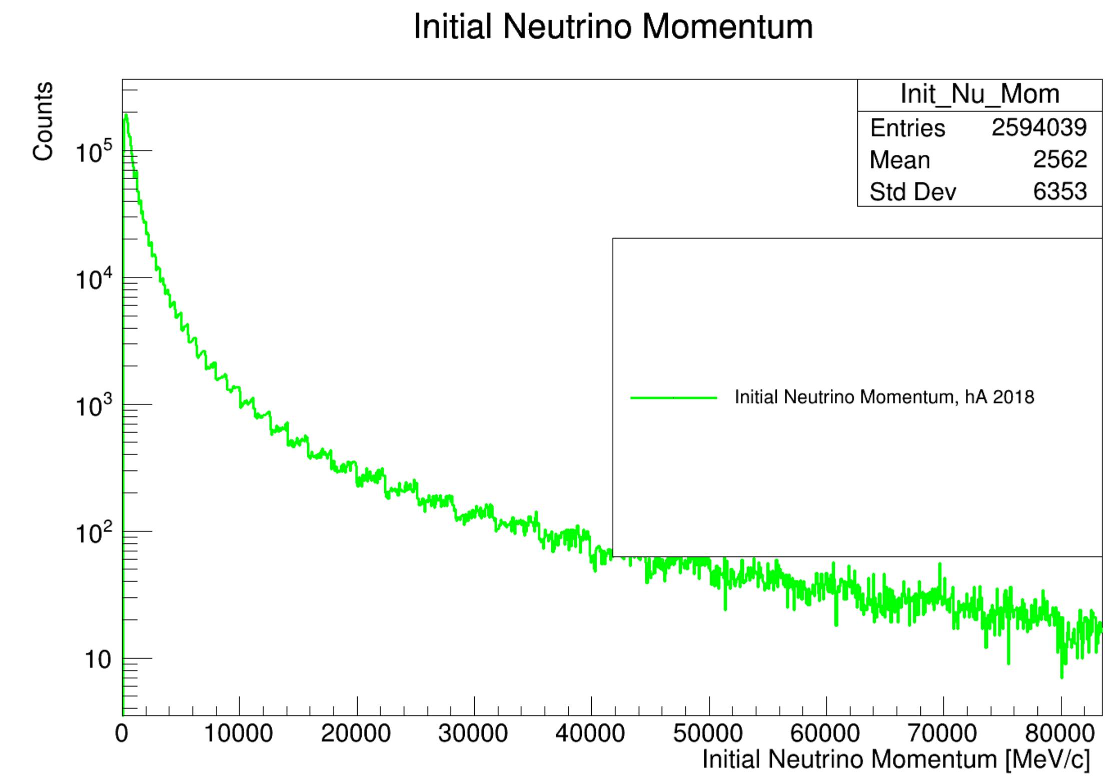
BEAM SIMULATION COMPARISONS TO COME

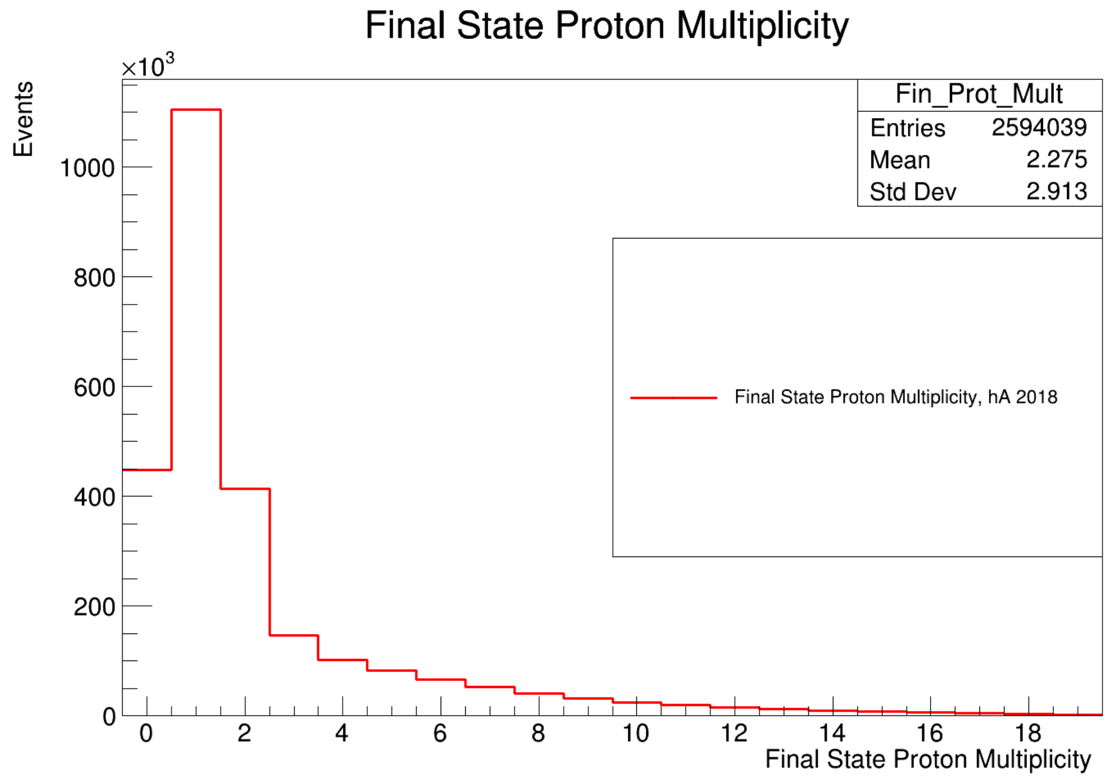
$E_{\nu} \epsilon (0.1,1)\,\mathrm{GeV}$, any ν flavor, any process, $NpN\pi$ topologies

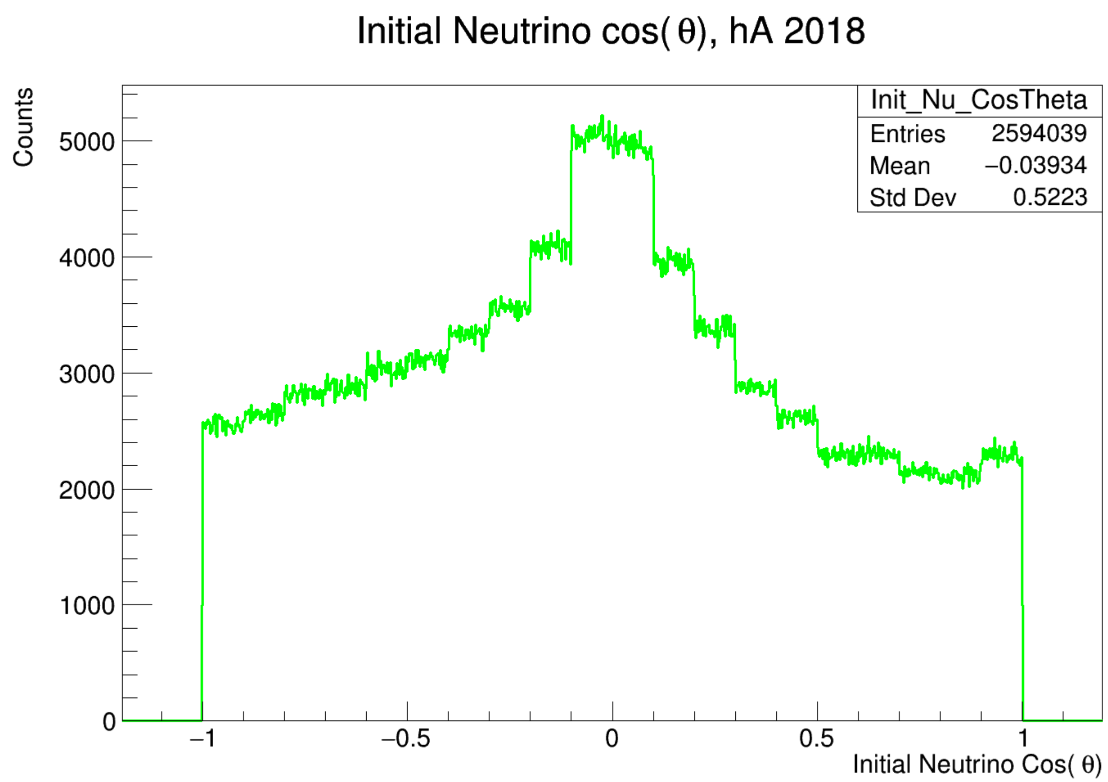
Using kinetic energy thresholds alone

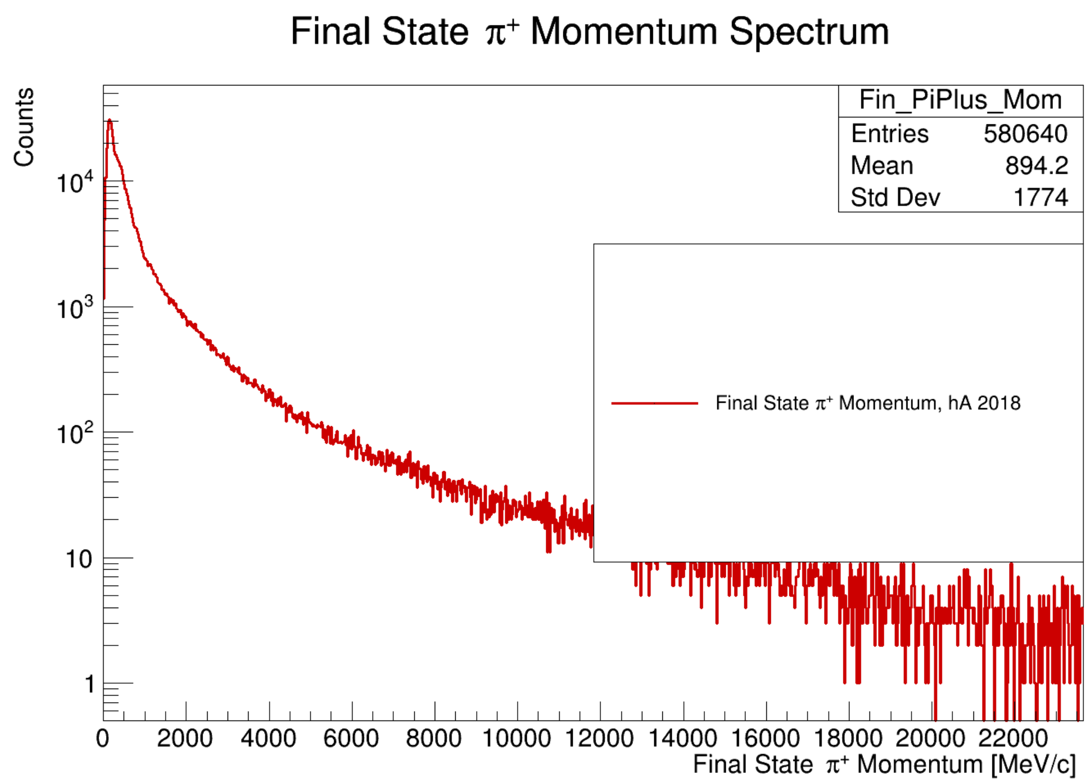
R. R. Richi

```
train indices size:    ~1.8M (~75%)
valid indices size:      ~200k (5%)
test  indices size:     ~500k (20%)
```









Example histograms

Full Honda atmospheric flux sample with oscillations at Homestake (same as in PDK study)

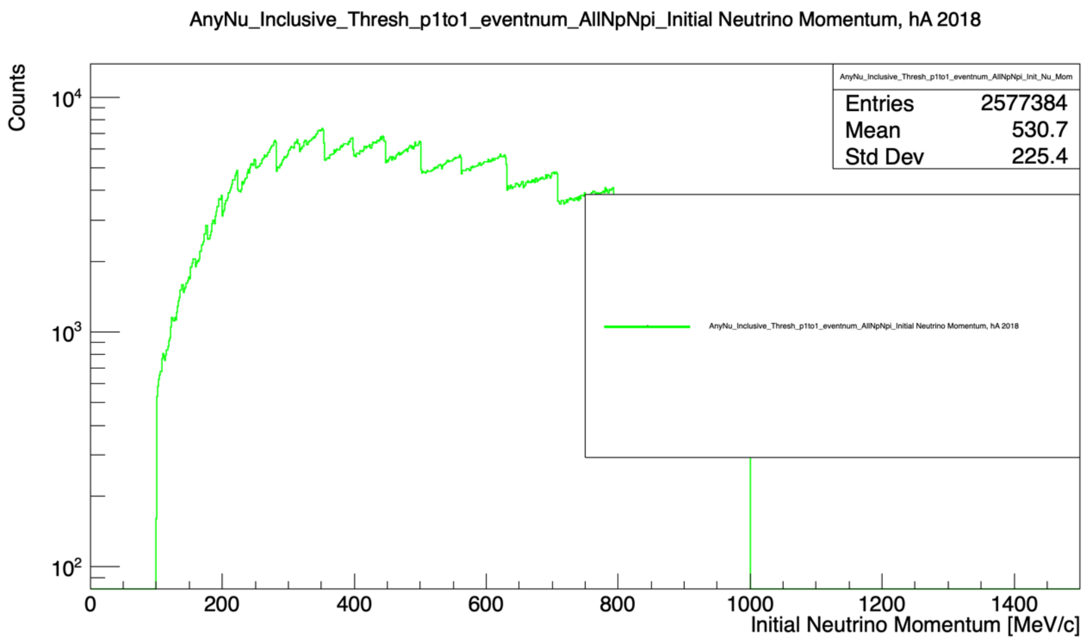
$E_\nu \epsilon (0.1,100)\mathrm{GeV}$

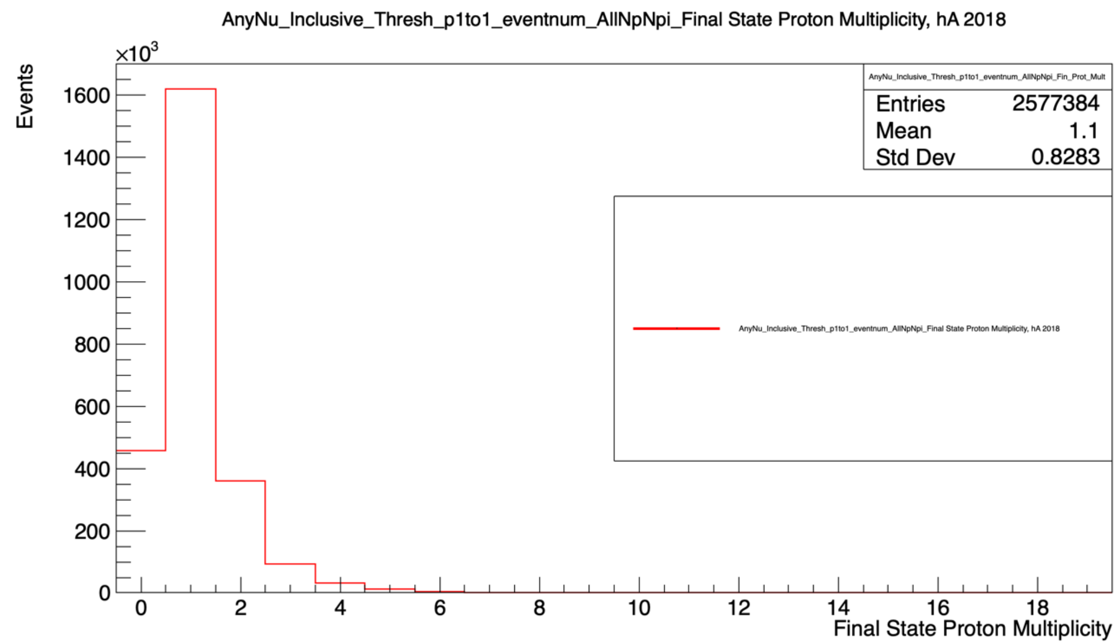
Any ν flavor

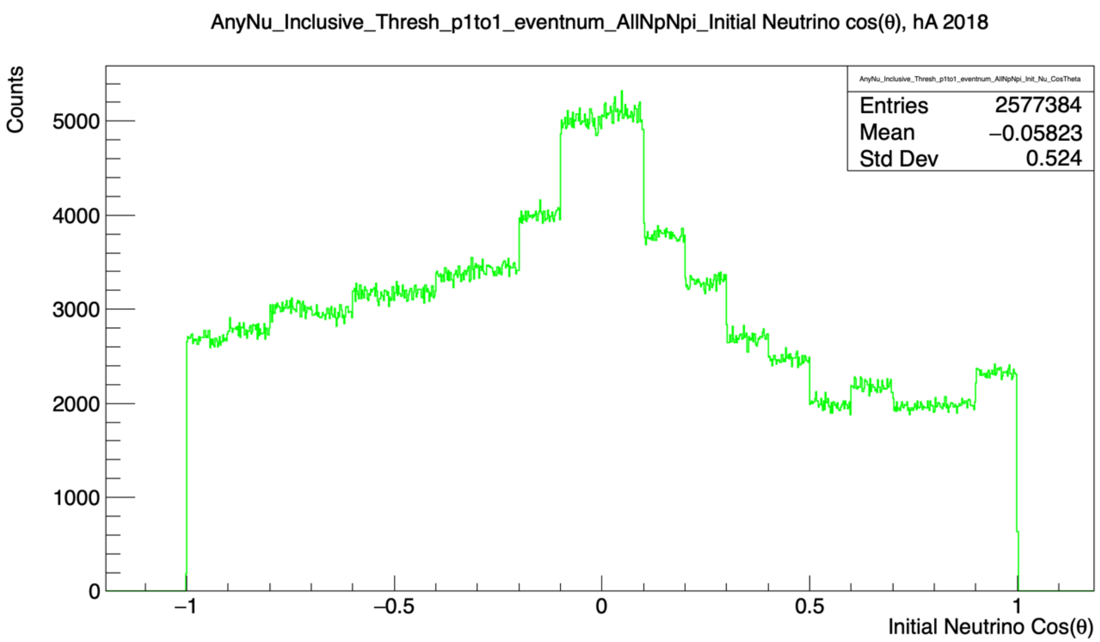
Any process

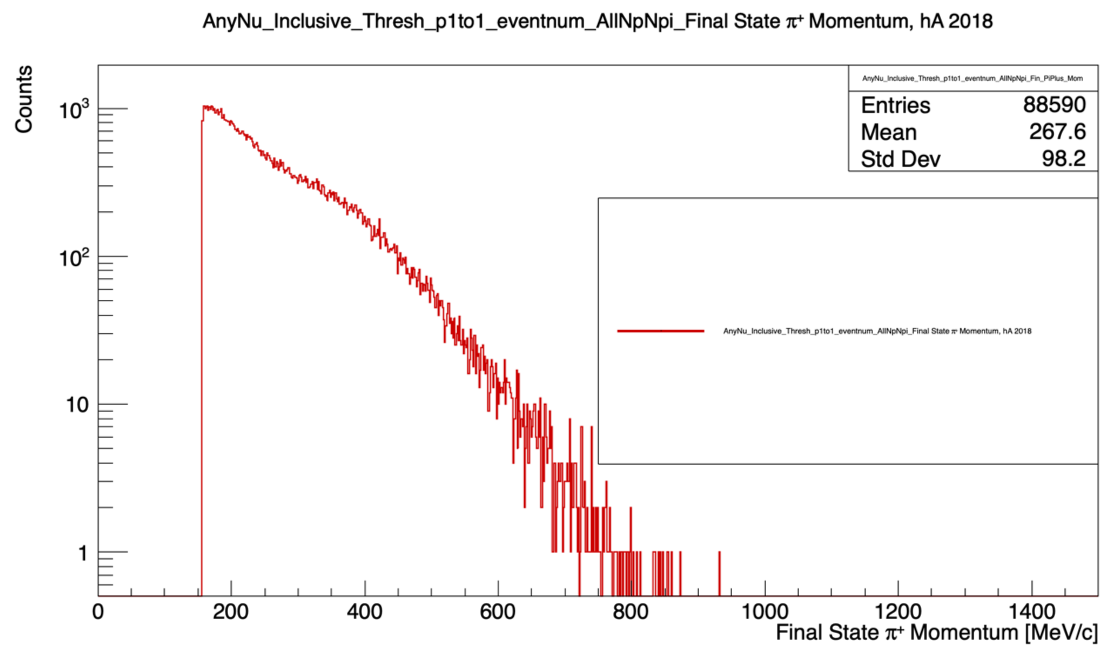
$NpN\pi$

hA_BR Nuclear model configuration









Example histograms

PARTIAL Honda atmospheric flux sample with oscillations at Homestake (same as in PDK study)

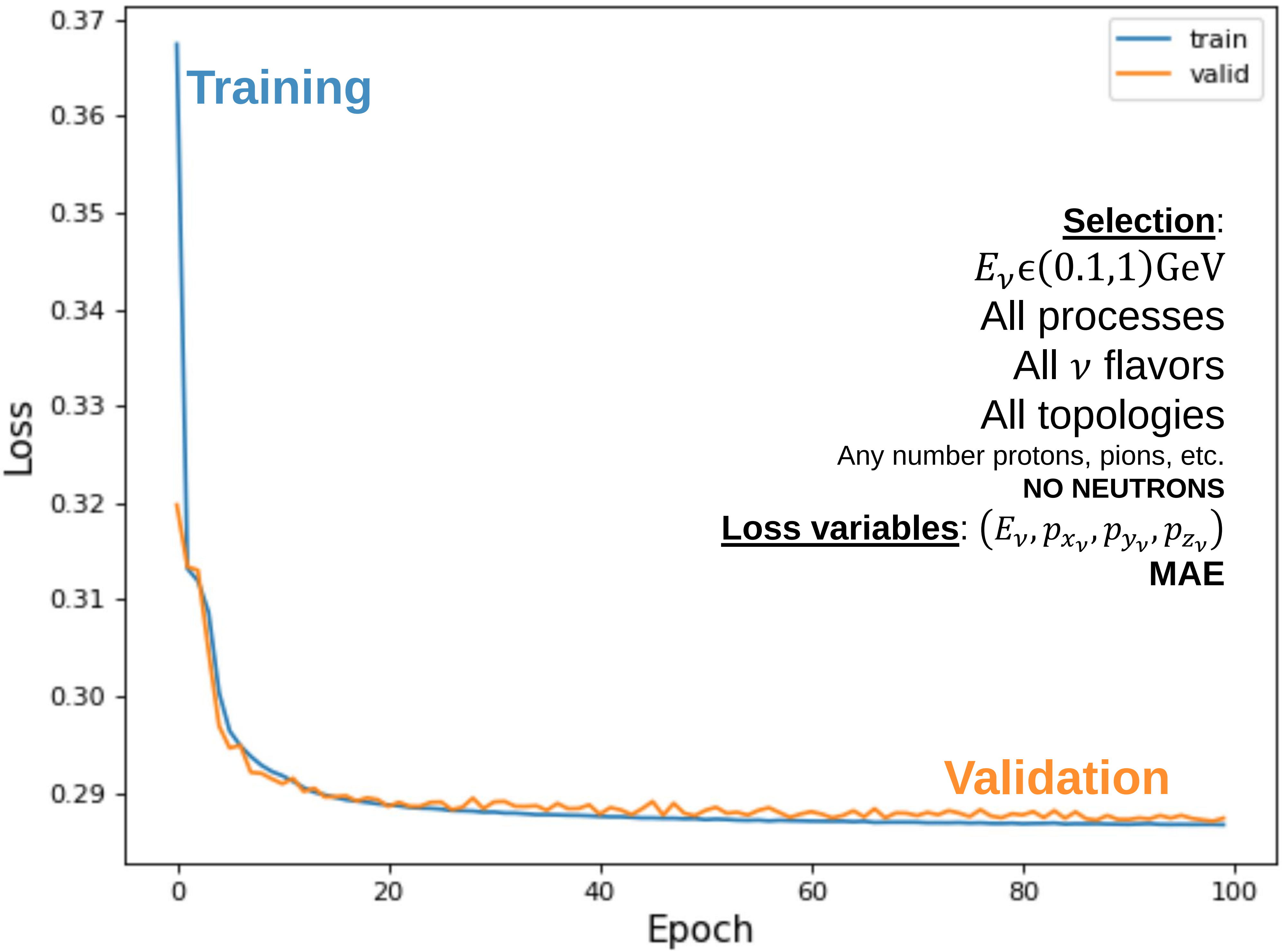
$E_{\nu} \epsilon (0.1,1)\mathrm{GeV}$

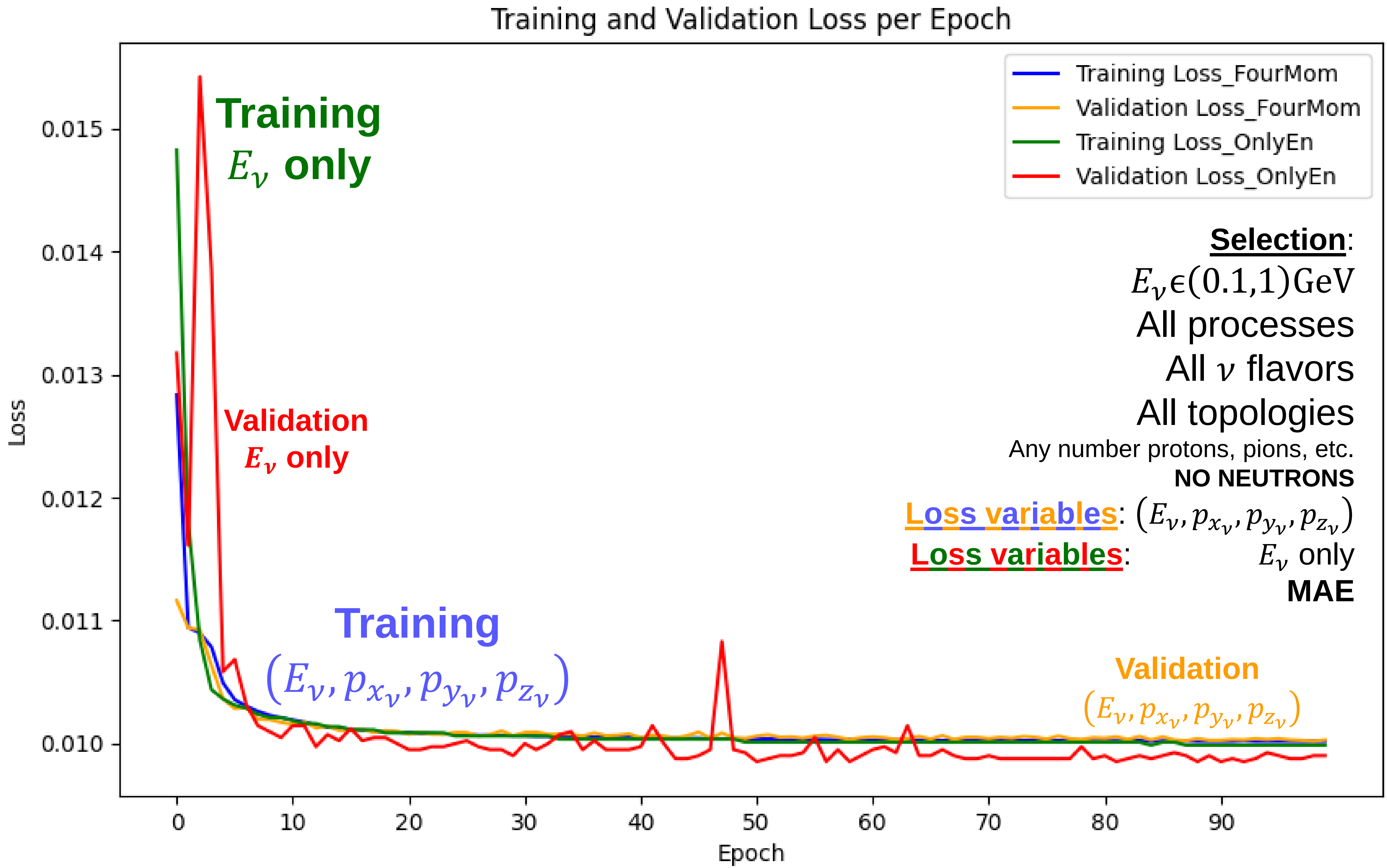
Any ν flavor

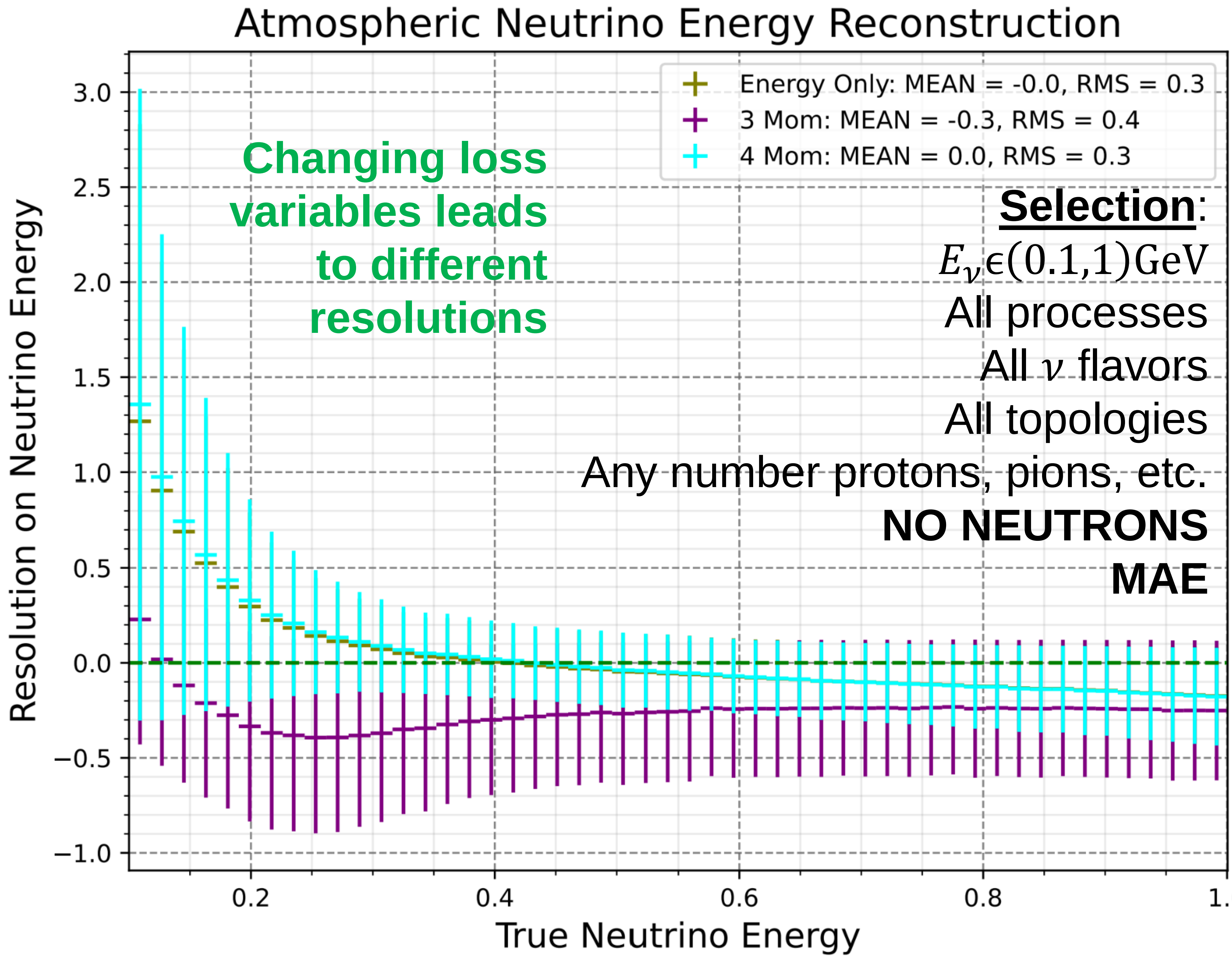
Any process

$NpN\pi$

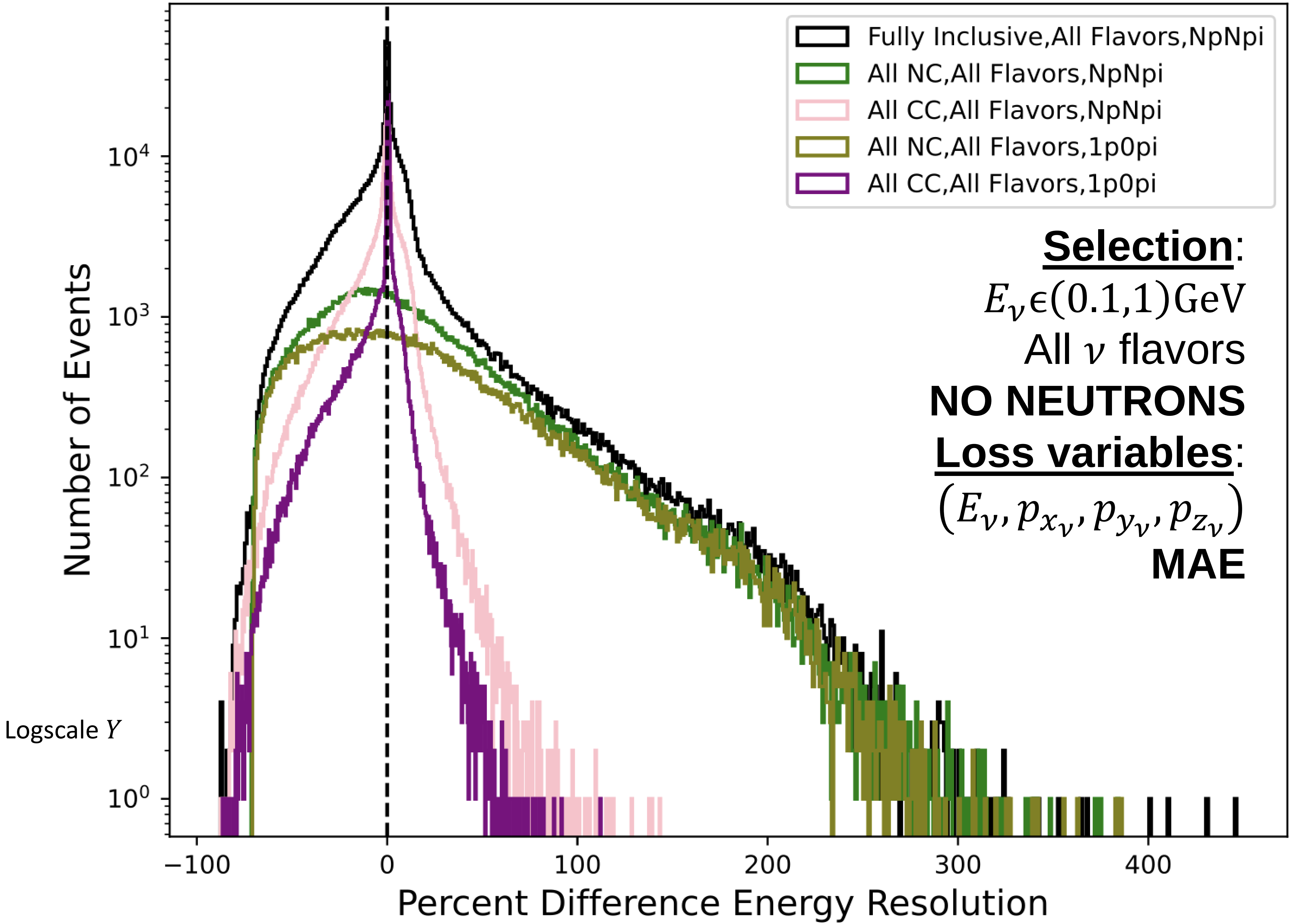
hA_BR Nuclear model configuration


Training
Validation
train
valid
Selection:
$E_\nu \epsilon (0.1, 1)\,\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE
Loss
Epoch
0.37
0.36
0.35
0.34
0.33
0.32
0.31
0.30
0.29
0
20
40
60
80
100


Training and Validation Loss per Epoch
Training Loss_FourMom
Validation Loss_FourMom
Training Loss_OnlyEn
Validation Loss_OnlyEn
Training
E_ν only
Validation
E_ν only
Training
$(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
Validation
$(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
Selection:
$E_\nu \epsilon (0.1, 1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
Loss variables: E_ν only
MAE
Loss
Epoch
0.015
0.014
0.013
0.012
0.011
0.010
0
10
20
30
40
50
60
70
80
90



Atmospheric Neutrino Energy Reconstruction
Energy Only: MEAN = -0.0, RMS = 0.3
3 Mom: MEAN = -0.3, RMS = 0.4
4 Mom: MEAN = 0.0, RMS = 0.3
Changing loss variables leads to different resolutions
Selection:
$E_\nu \epsilon (0.1,1)$GeV
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
MAE
Resolution on Neutrino Energy
True Neutrino Energy
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0.2
0.4
0.6
0.8
1.0


Fully Inclusive,All Flavors,NpNpi
All NC,All Flavors,NpNpi
All CC,All Flavors,NpNpi
All NC,All Flavors,1p0pi
All CC,All Flavors,1p0pi
Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All ν flavors
NO NEUTRONS
Loss variables:
$\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$
MAE
Number of Events
10^4
10^3
10^2
10^1
10^0
−100
0
100
200
300
400
Percent Difference Energy Resolution
Logscale Y

Selection:

$E_\nu \epsilon (0.1,1)\mathrm{GeV}$

All processes

All ν flavors

All topologies

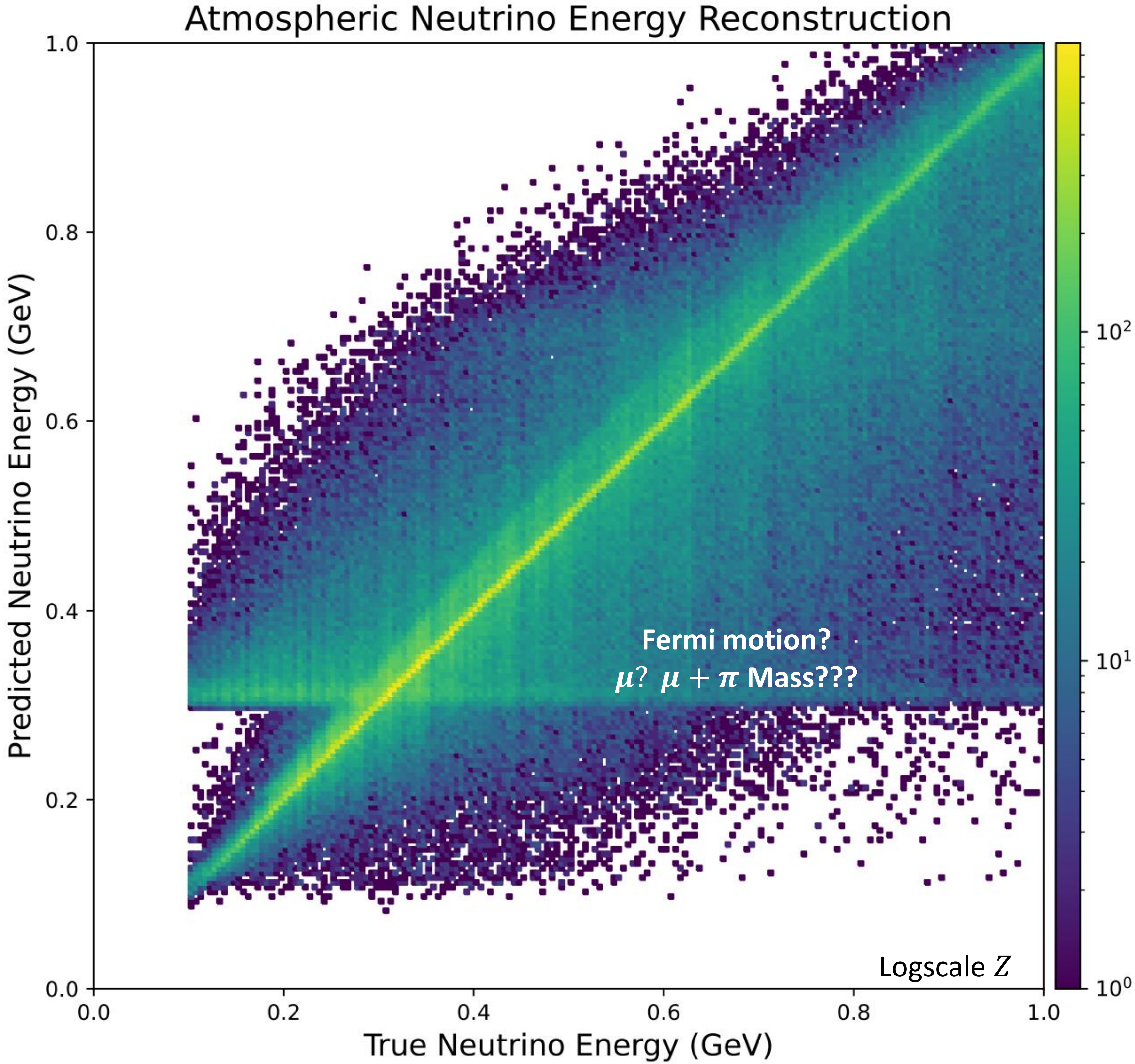
Any number protons, pions, etc.

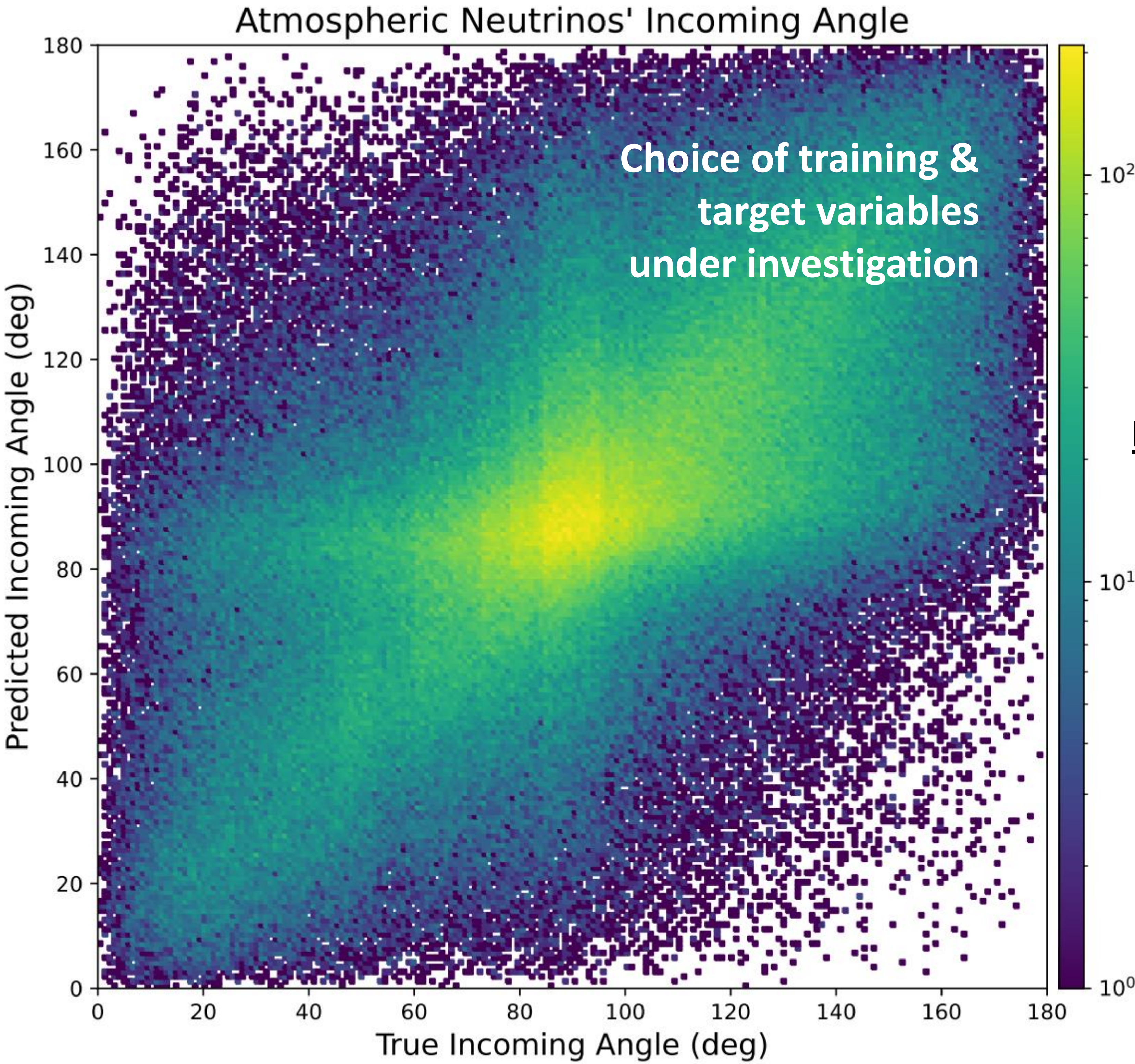
NO NEUTRONS

Loss variables:

$\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

MAE

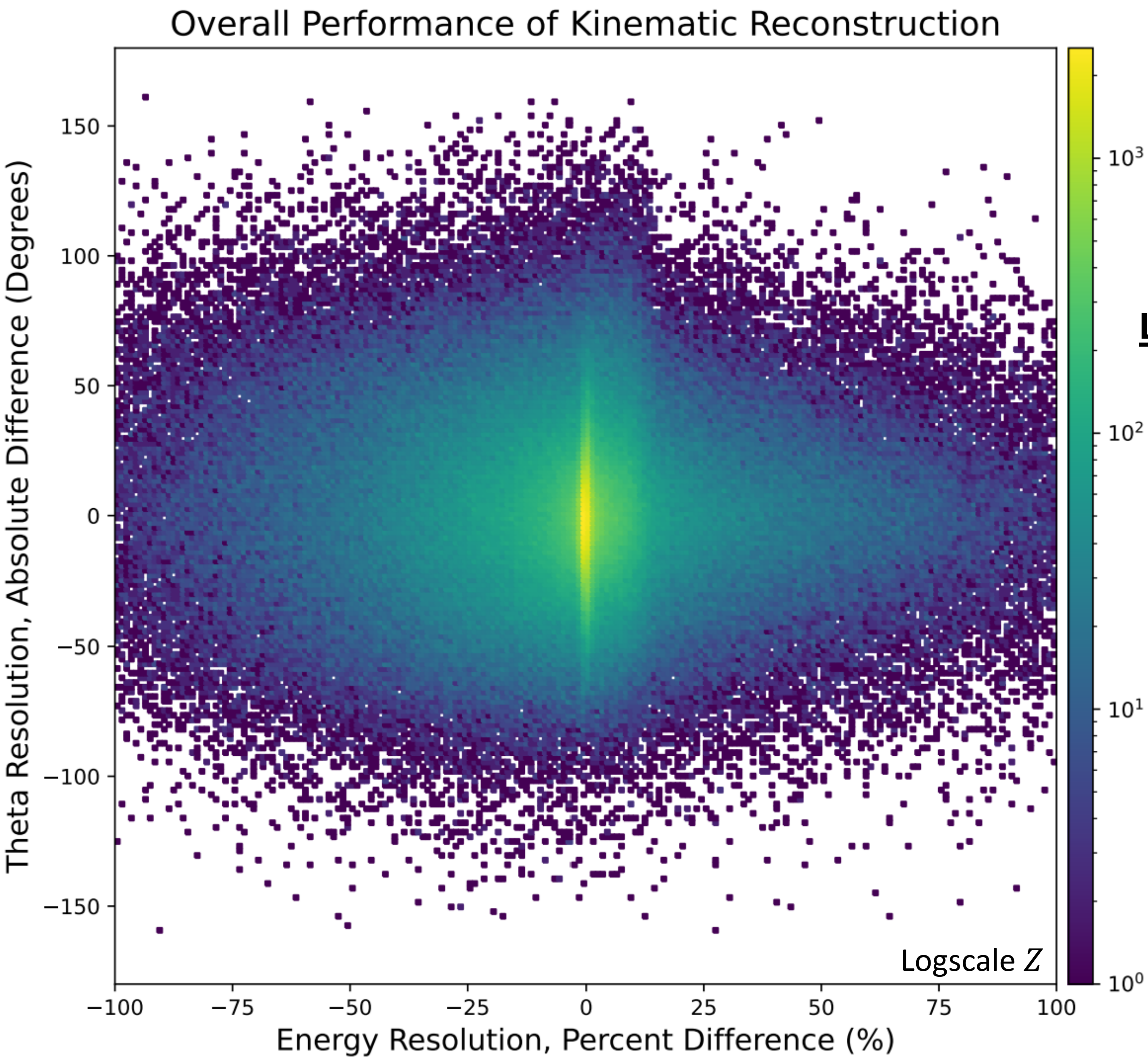




Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS

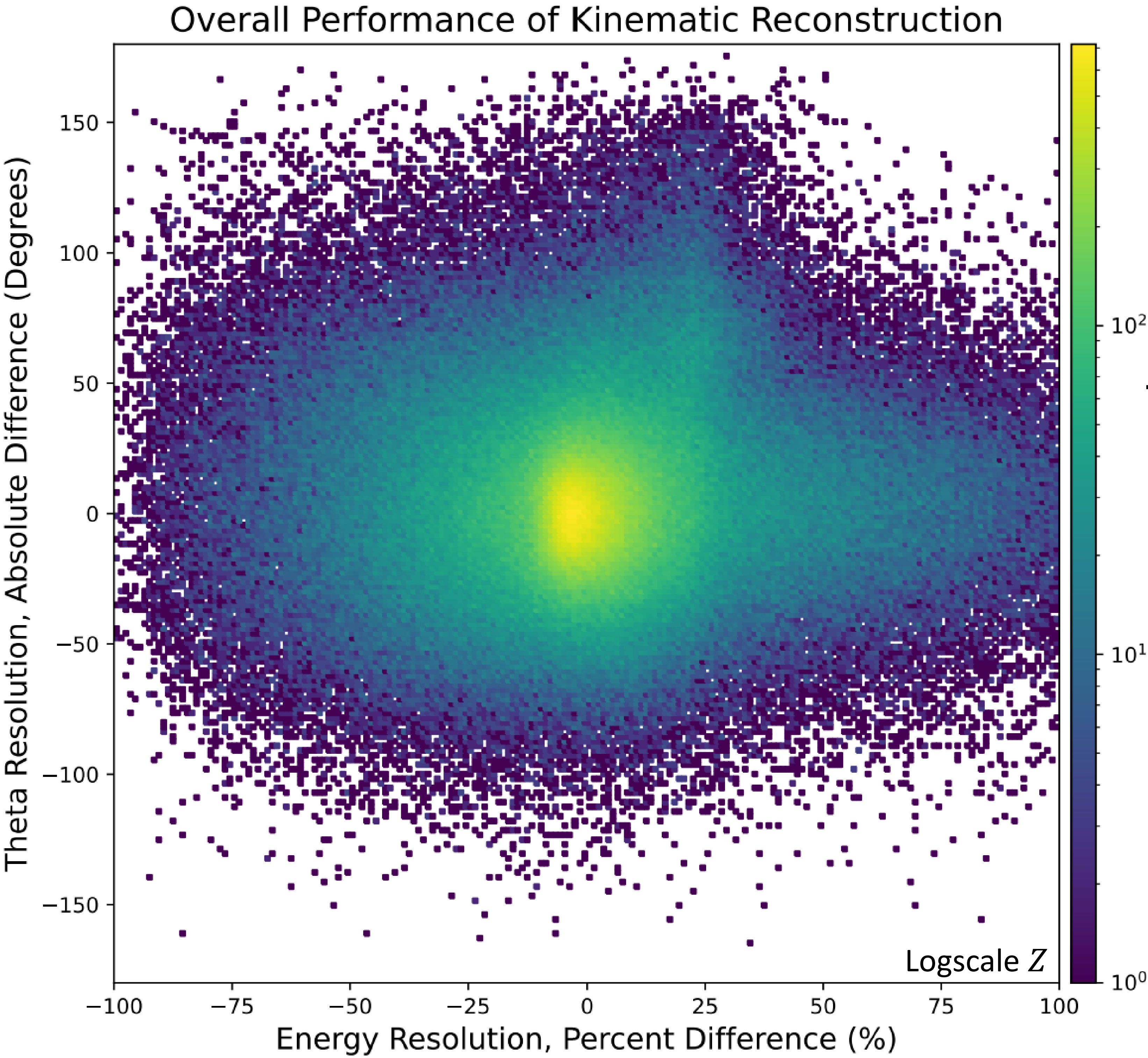
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE

Logscale Z

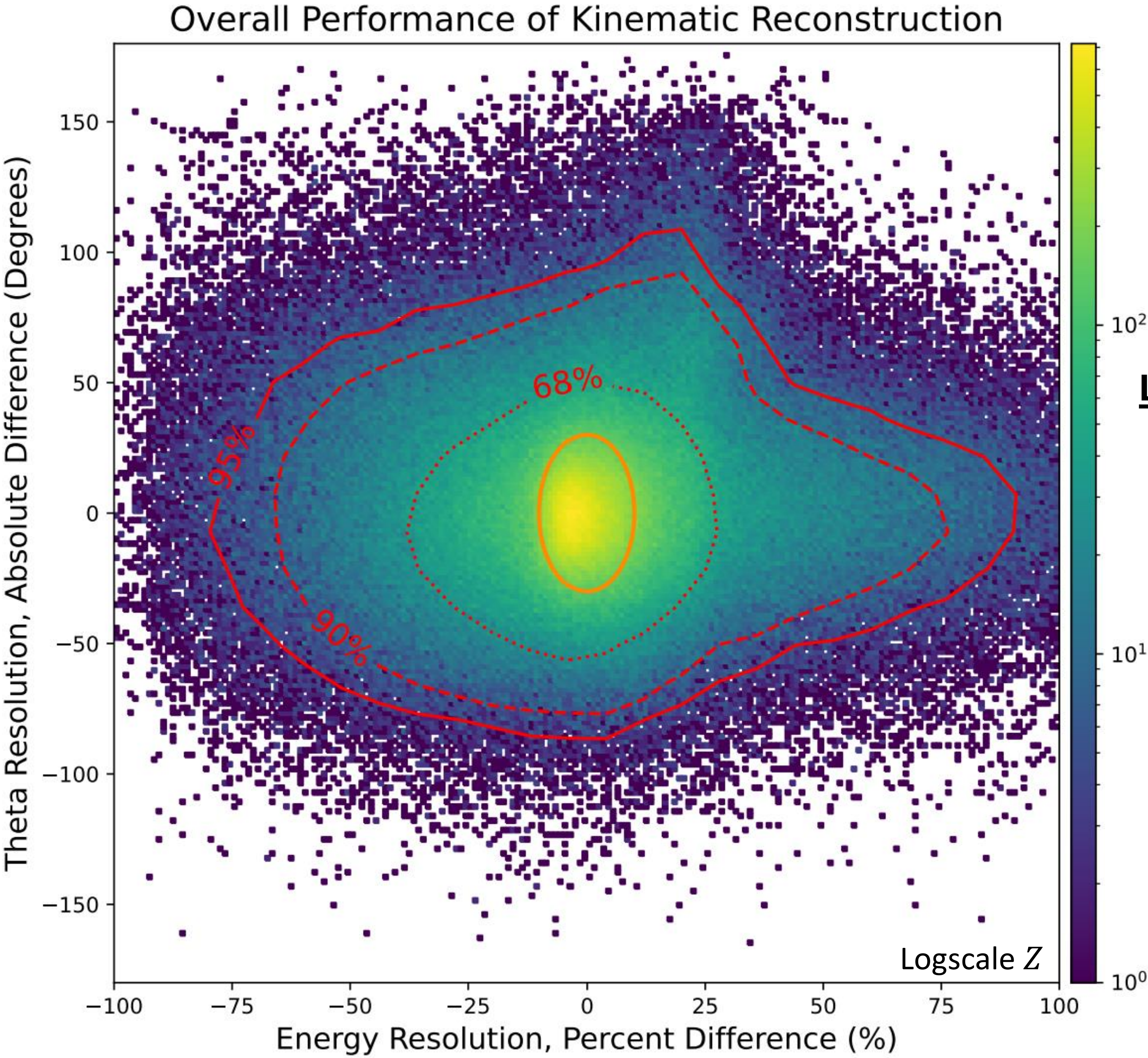


Selection:
$E_\nu \epsilon (0.1, 1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS

Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE

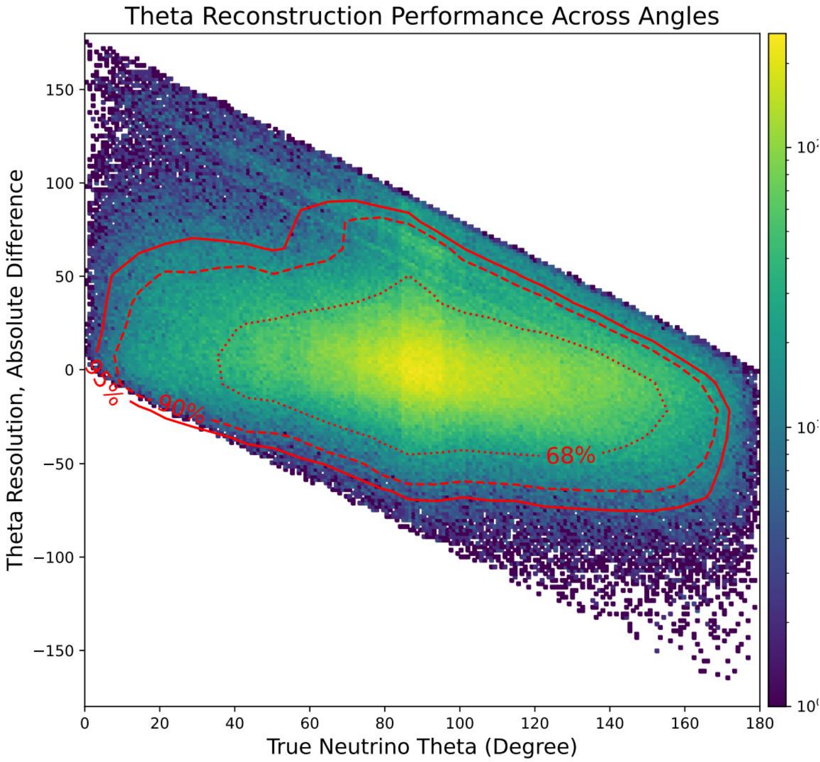


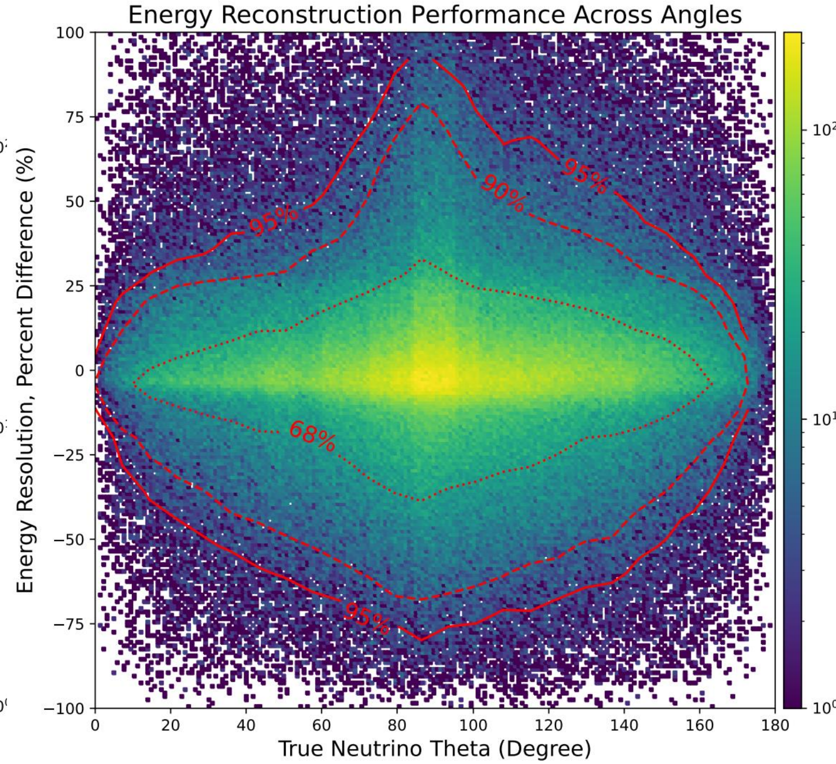
Selection:
$E_\nu \epsilon (0.1, 1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$
MSE



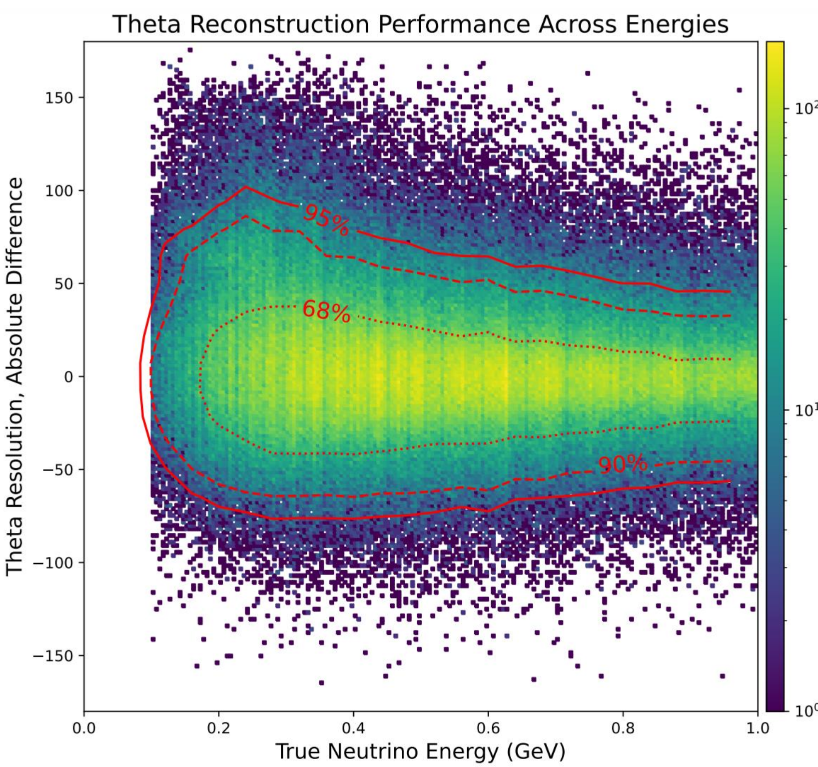
Selection:
$E_{\nu} \epsilon (0.1,1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS

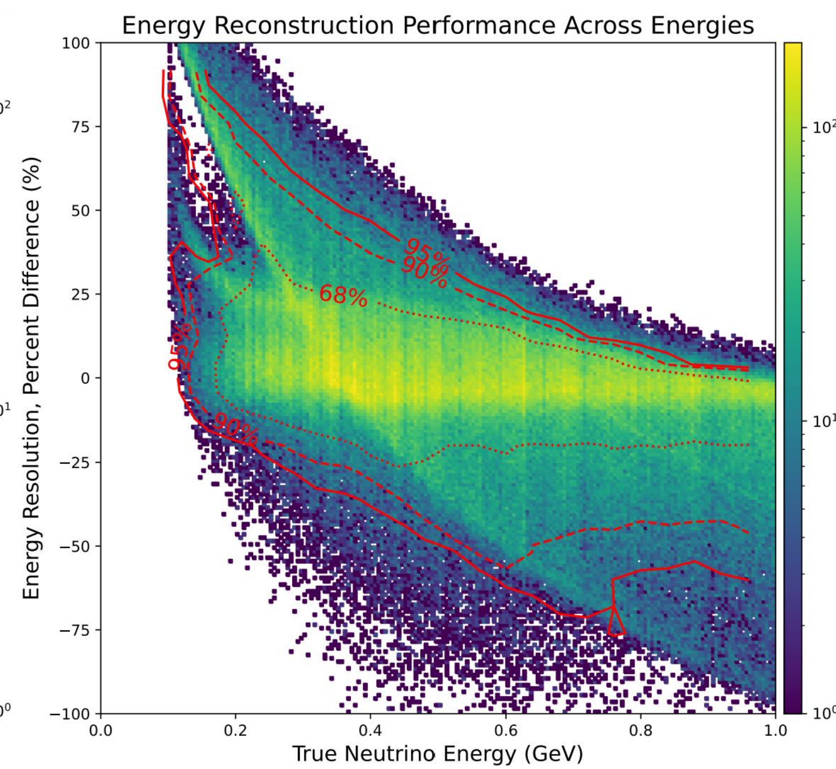
Loss variables: $\left(E_{\nu}, p_{x_{\nu}}, p_{y_{\nu}}, p_{z_{\nu}}\right)$
MSE

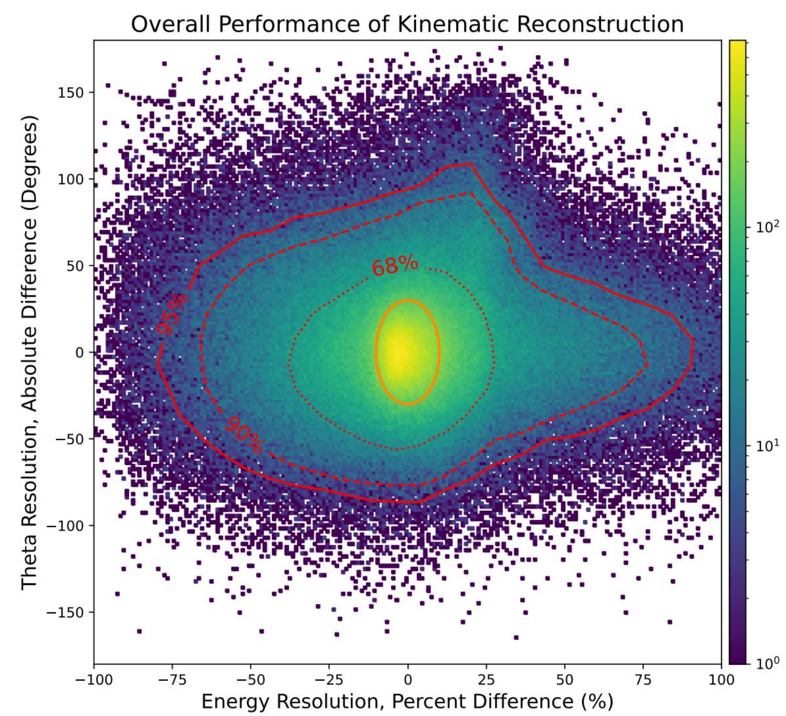


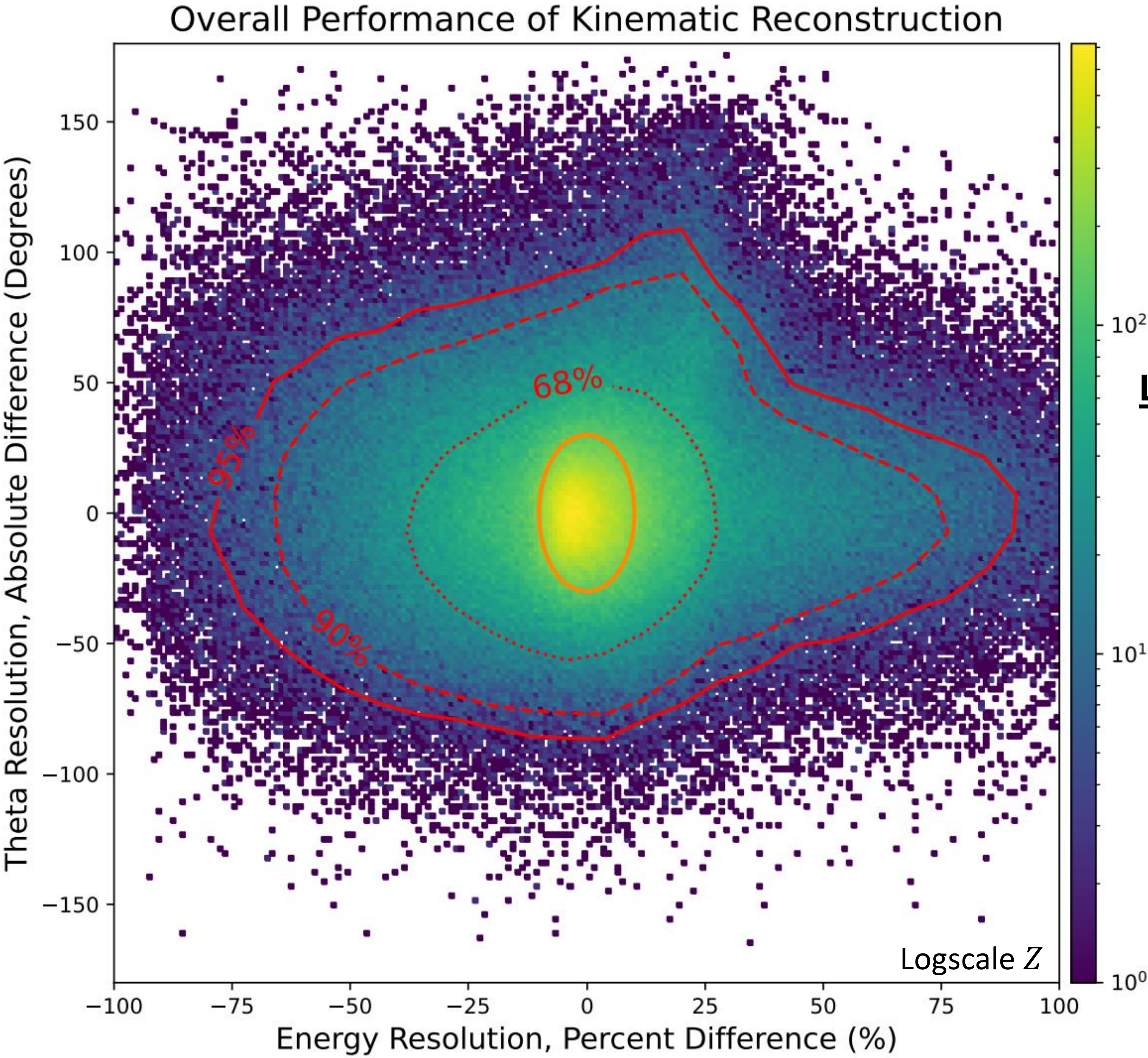


Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$
****MSE****









<u>Selection</u>:

$E_\nu \epsilon (0.1,1)\mathrm{GeV}$

All processes

All ν flavors

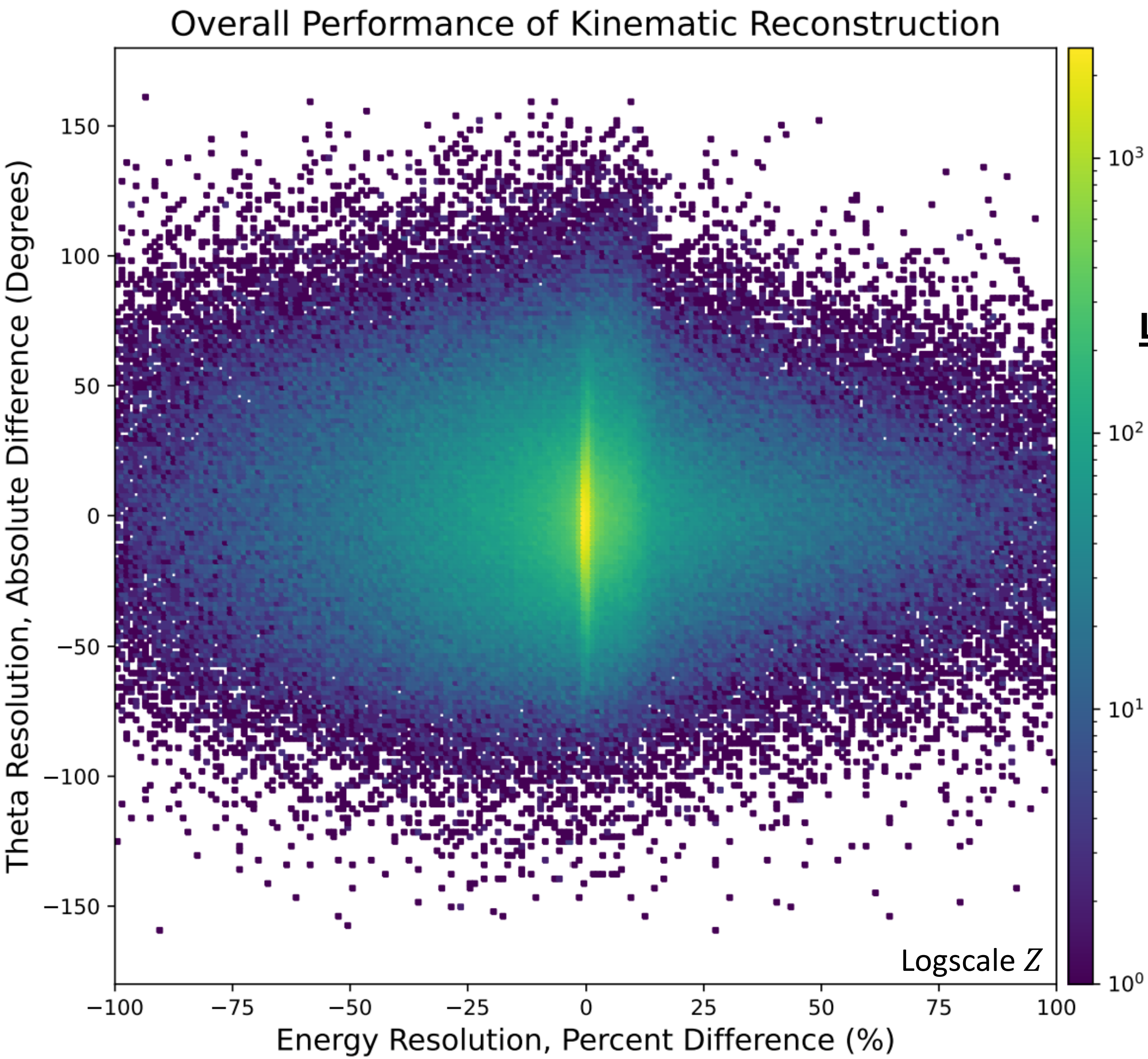
All topologies

Any number protons, pions, etc.

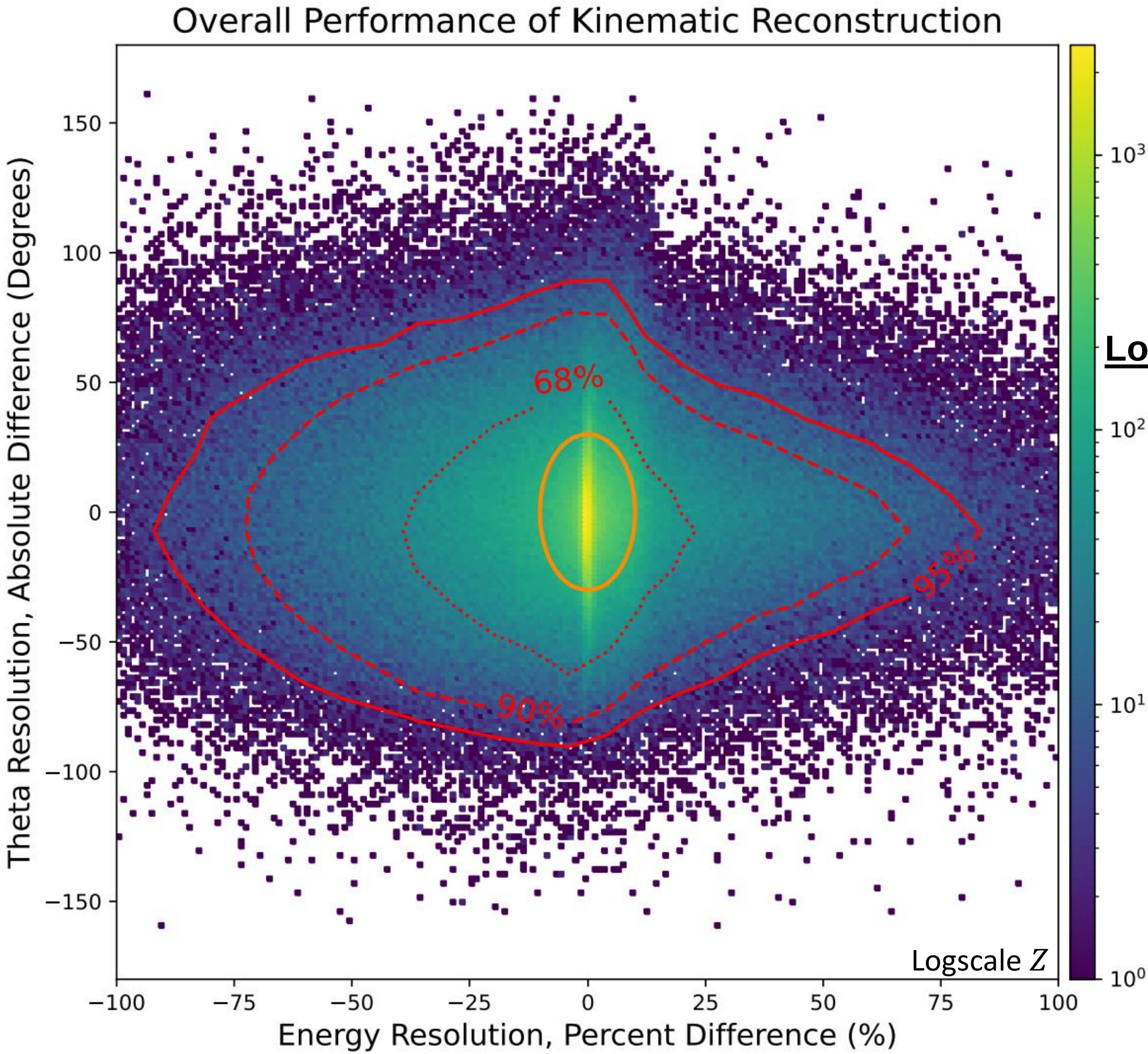
NO NEUTRONS

<u>Loss variables</u>: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

****MSE****

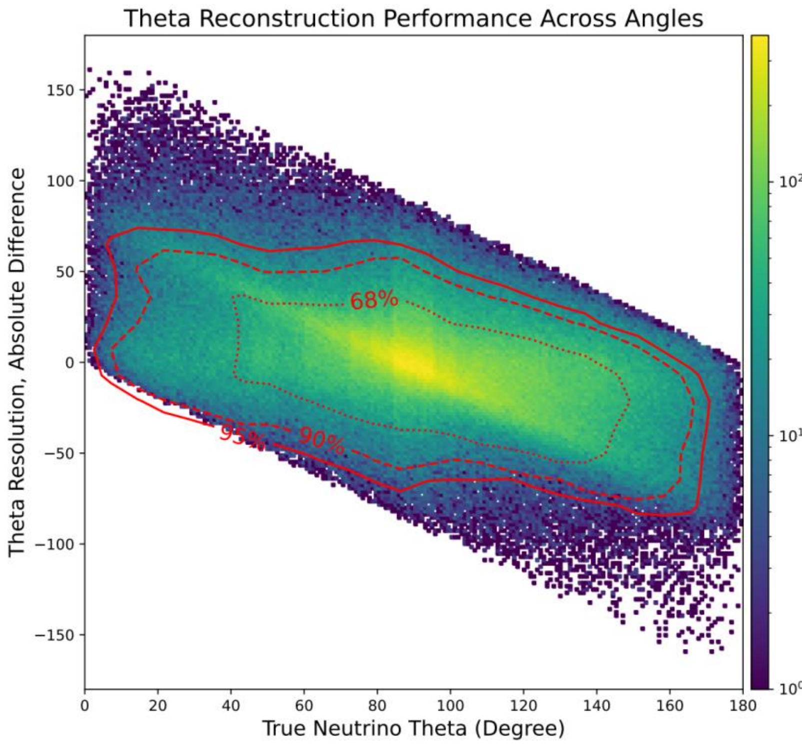


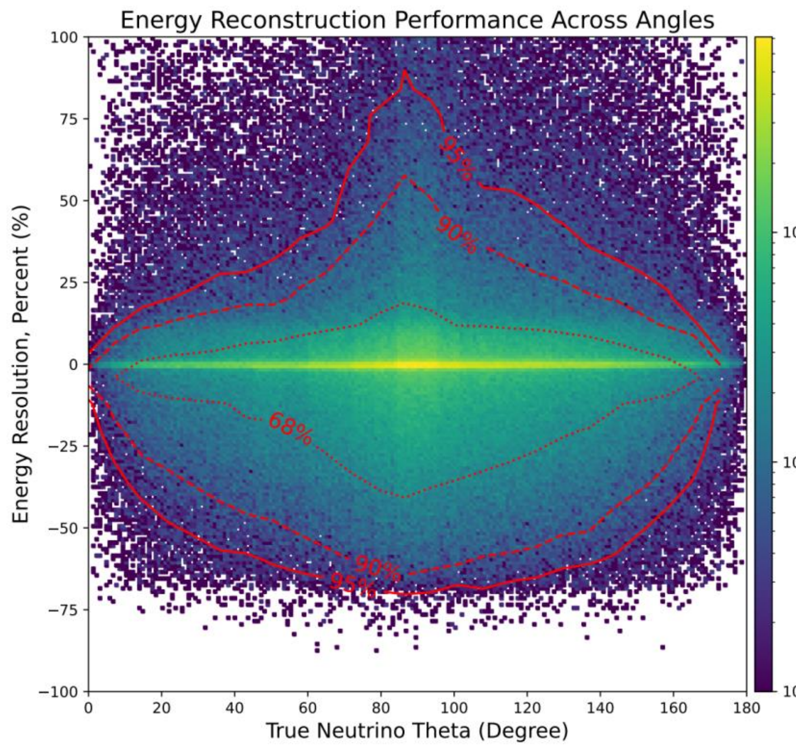
Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$
MAE



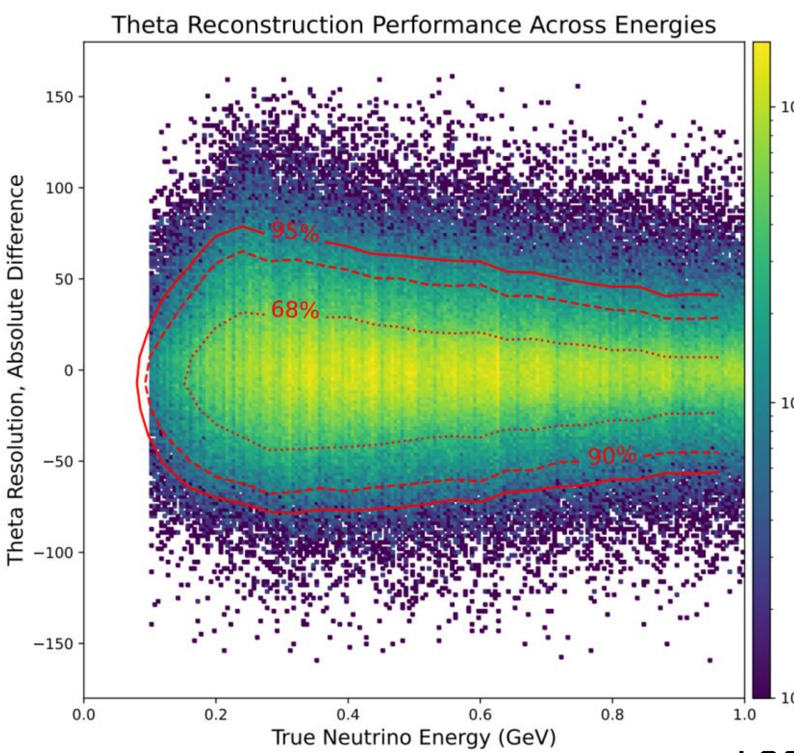
Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS

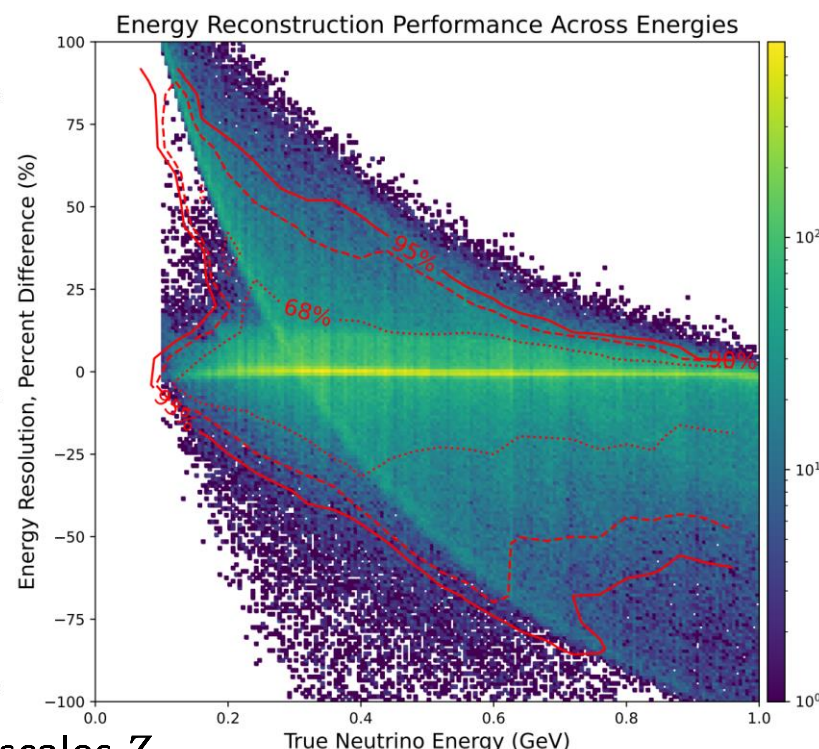
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE

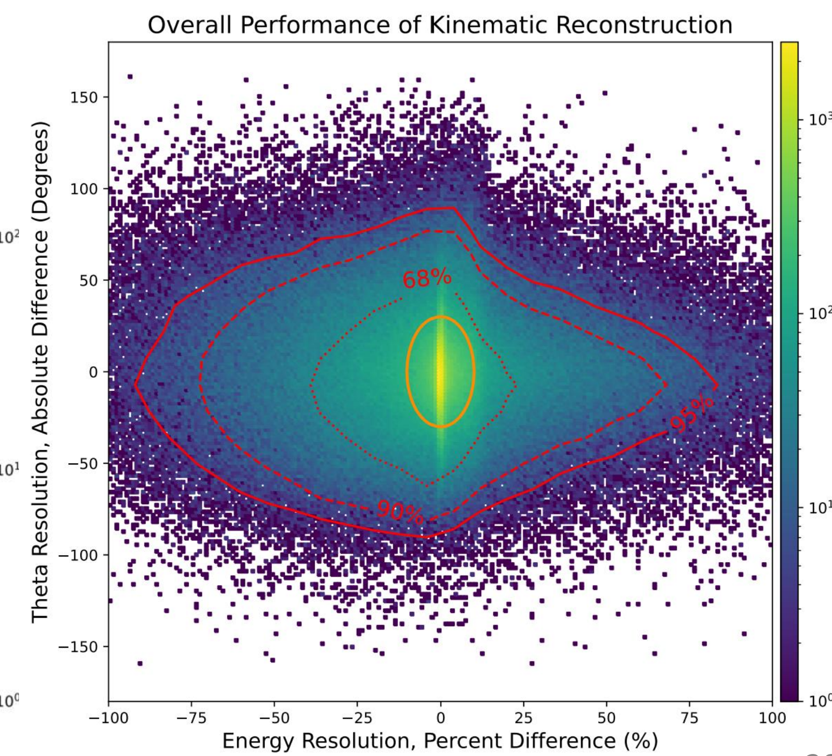




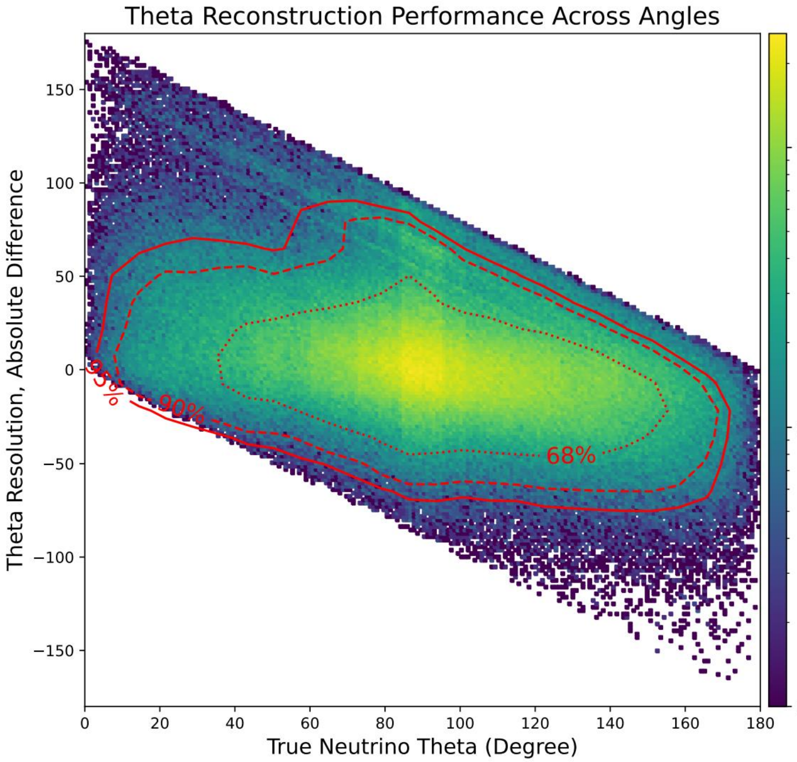
Selection:
$E_\nu \epsilon (0.1, 1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE

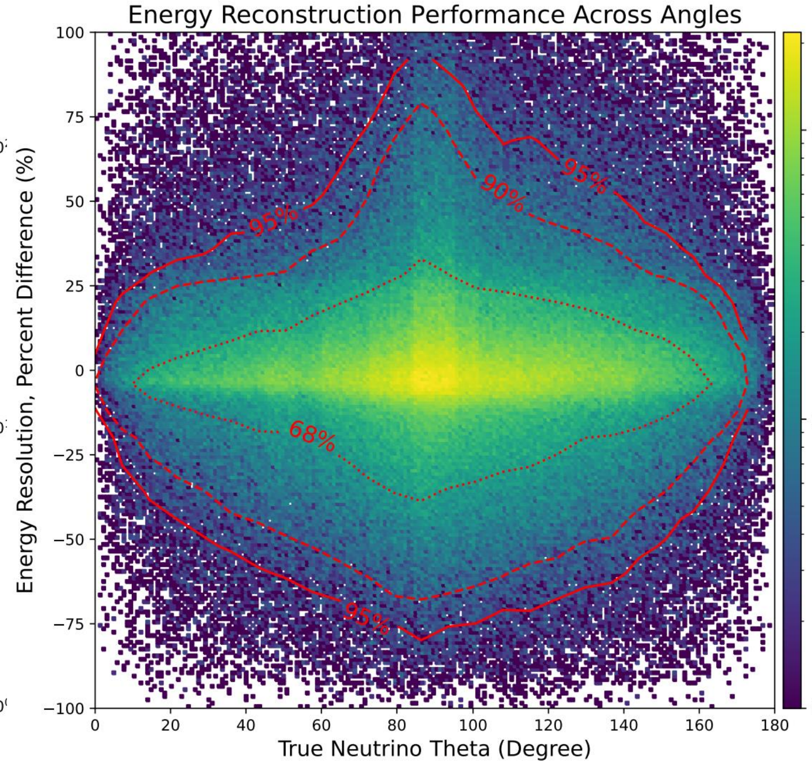




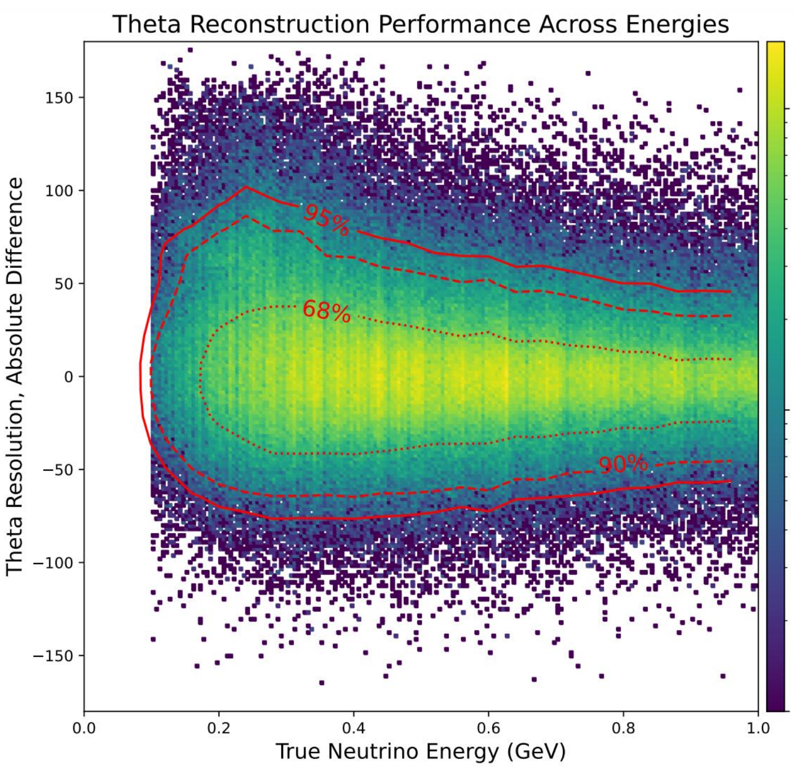


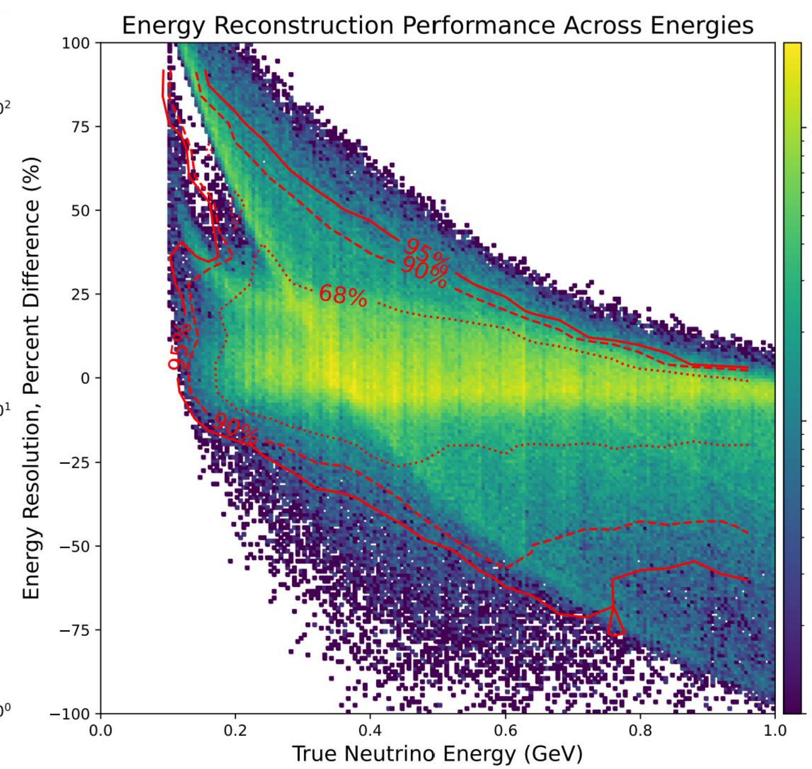
Logscales Z

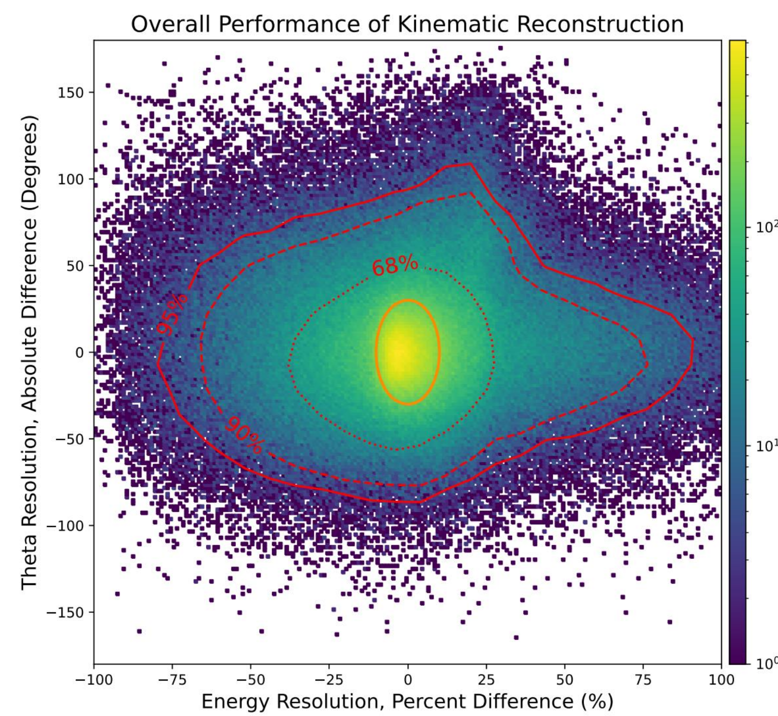




Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All processes
All ν flavors
All topologies
Any number protons, pions, etc.
NO NEUTRONS
Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$
****MSE****







<u>Selection</u>:

TRAINING ON $E_\nu \epsilon (0.1,100)\mathrm{GeV}$

SHOWING ONLY $E_\nu \epsilon (0.1,1)\mathrm{GeV}$

****NO THRESHOLDS****

All processes

All ν flavors

All topologies

Any number protons, pions, etc.

NO NEUTRONS

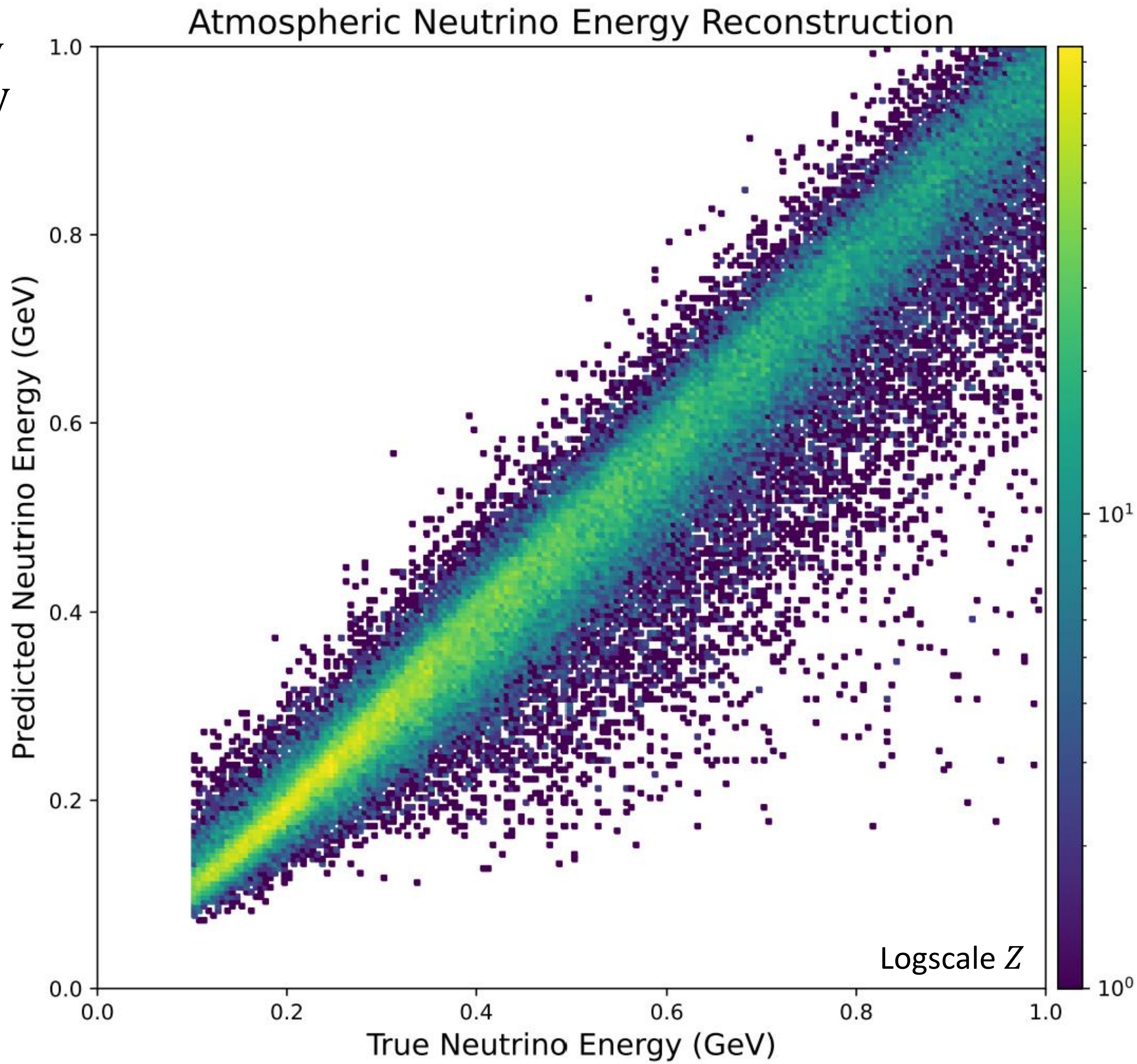
<u>Loss variables</u>: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

****MAE****

PREVIOUS TRAINING VARIABLES

****CHEATS ON NC EVENTS!!****

```
"vector": [
    "Final_State_Particles_PDG",
    "Final_State_Particles_Mass",
    "Final_State_Particles_Energy",
    "Final_State_Particles_Momentum_X",
    "Final_State_Particles_Momentum_Y",
    "Final_State_Particles_Momentum_Z"

],
"scalar": [
    "Initial_State_Neutrino_PDG",
    "Final_State_Lepton_PDG",
    "Final_State_Lepton_Mass",
    "Final_State_Lepton_Energy",
    "Final_State_Lepton_Momentum_X",
    "Final_State_Lepton_Momentum_Y",
    "Final_State_Lepton_Momentum_Z"

],
"target": [
    "Initial_State_Neutrino_Energy",
    "Initial_State_Neutrino_Momentum_X",
    "Initial_State_Neutrino_Momentum_Y",
    "Initial_State_Neutrino_Momentum_Z"

],
```



Only showing ~100k events here!

<u>Selection</u>:
TRAINING ON $E_\nu \epsilon (0.1,100)\mathrm{GeV}$
SHOWING ONLY $E_\nu \epsilon (0.1,1)\mathrm{GeV}$
****NO THRESHOLDS****

All processes

All ν flavors

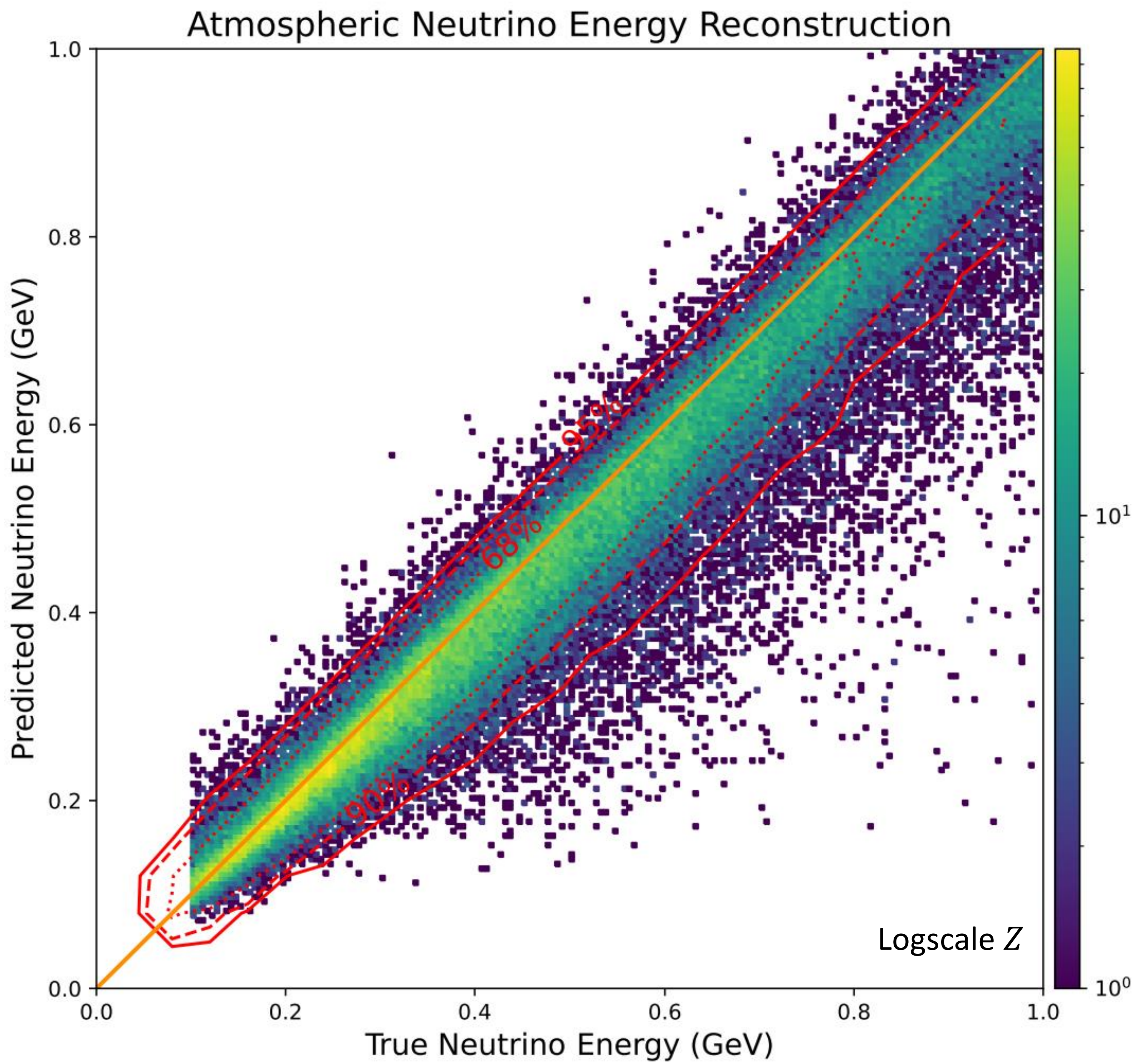
All topologies

Any number protons, pions, etc.
NO NEUTRONS

<u>Loss variables</u>: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

****MAE****
PREVIOUS TRAINING VARIABLES
****CHEATS ON NC EVENTS!!****

```
"vector": [
    "Final_State_Particles_PDG",
    "Final_State_Particles_Mass",
    "Final_State_Particles_Energy",
    "Final_State_Particles_Momentum_X",
    "Final_State_Particles_Momentum_Y",
    "Final_State_Particles_Momentum_Z"

],
"scalar": [
    "Initial_State_Neutrino_PDG",
    "Final_State_Lepton_PDG",
    "Final_State_Lepton_Mass",
    "Final_State_Lepton_Energy",
    "Final_State_Lepton_Momentum_X",
    "Final_State_Lepton_Momentum_Y",
    "Final_State_Lepton_Momentum_Z"

],
"target": [
    "Initial_State_Neutrino_Energy",
    "Initial_State_Neutrino_Momentum_X",
    "Initial_State_Neutrino_Momentum_Y",
    "Initial_State_Neutrino_Momentum_Z"

],
```



Only showing ~100k events here!

Selection:

TRAINING ON $E_\nu \epsilon (0.1,100)\mathrm{GeV}$

SHOWING ONLY $E_\nu \epsilon (0.1,1)\mathrm{GeV}$

****NO THRESHOLDS****

All processes

All ν flavors

All topologies

Any number protons, pions, etc.

NO NEUTRONS

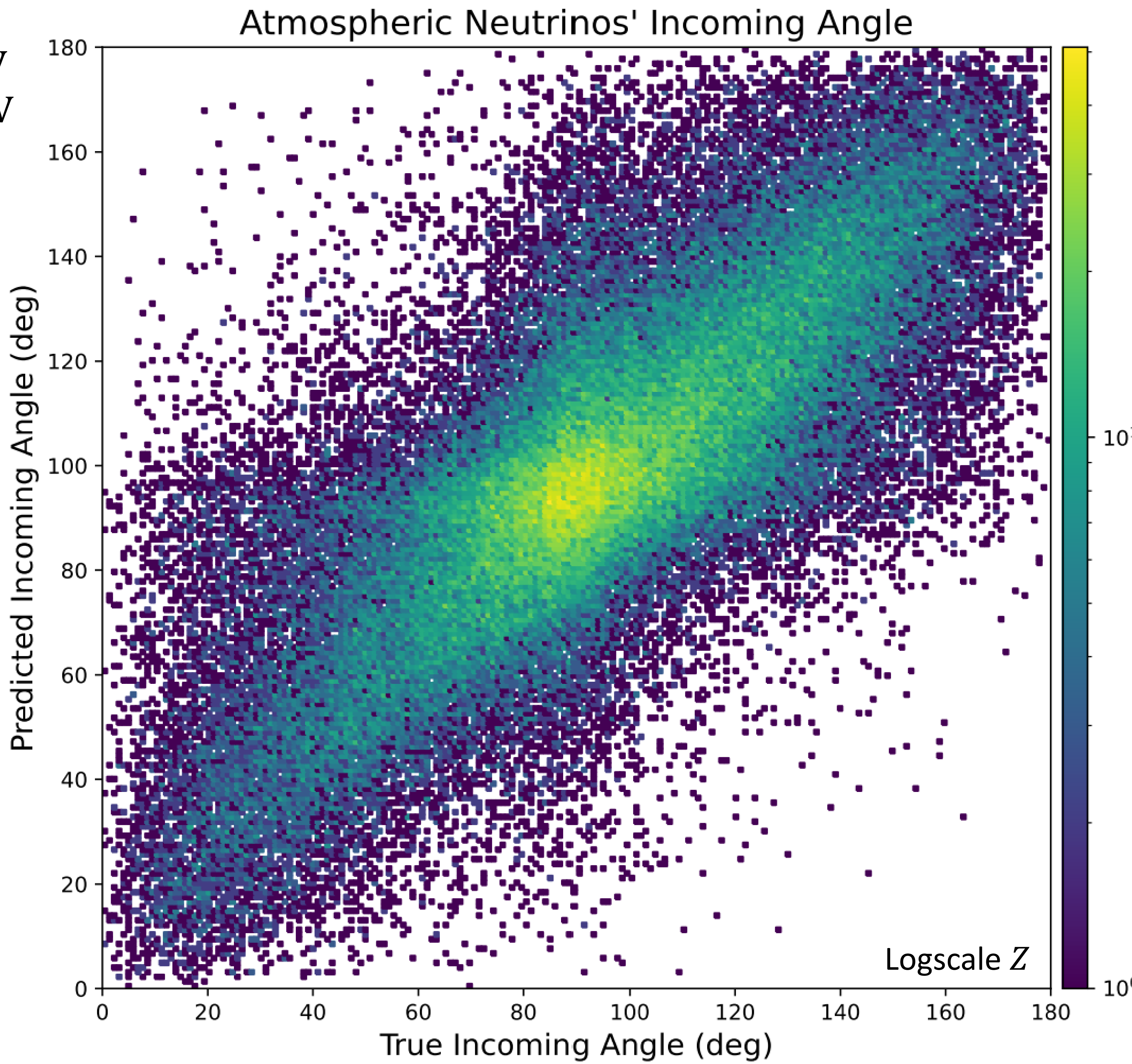
Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

****MAE****

PREVIOUS TRAINING VARIABLES

****CHEATS ON NC EVENTS!!****

```
"vector": [
    "Final_State_Particles_PDG",
    "Final_State_Particles_Mass",
    "Final_State_Particles_Energy",
    "Final_State_Particles_Momentum_X",
    "Final_State_Particles_Momentum_Y",
    "Final_State_Particles_Momentum_Z"

],
"scalar": [
    "Initial_State_Neutrino_PDG",
    "Final_State_Lepton_PDG",
    "Final_State_Lepton_Mass",
    "Final_State_Lepton_Energy",
    "Final_State_Lepton_Momentum_X",
    "Final_State_Lepton_Momentum_Y",
    "Final_State_Lepton_Momentum_Z"

],
"target": [
    "Initial_State_Neutrino_Energy",
    "Initial_State_Neutrino_Momentum_X",
    "Initial_State_Neutrino_Momentum_Y",
    "Initial_State_Neutrino_Momentum_Z"

],
```



Only showing ~100k events here!

Selection:

TRAINING ON $E_\nu \epsilon (0.1,100)\mathrm{GeV}$

SHOWING ONLY $E_\nu \epsilon (0.1,1)\mathrm{GeV}$

****NO THRESHOLDS****

All processes

All ν flavors

All topologies

Any number protons, pions, etc.

NO NEUTRONS

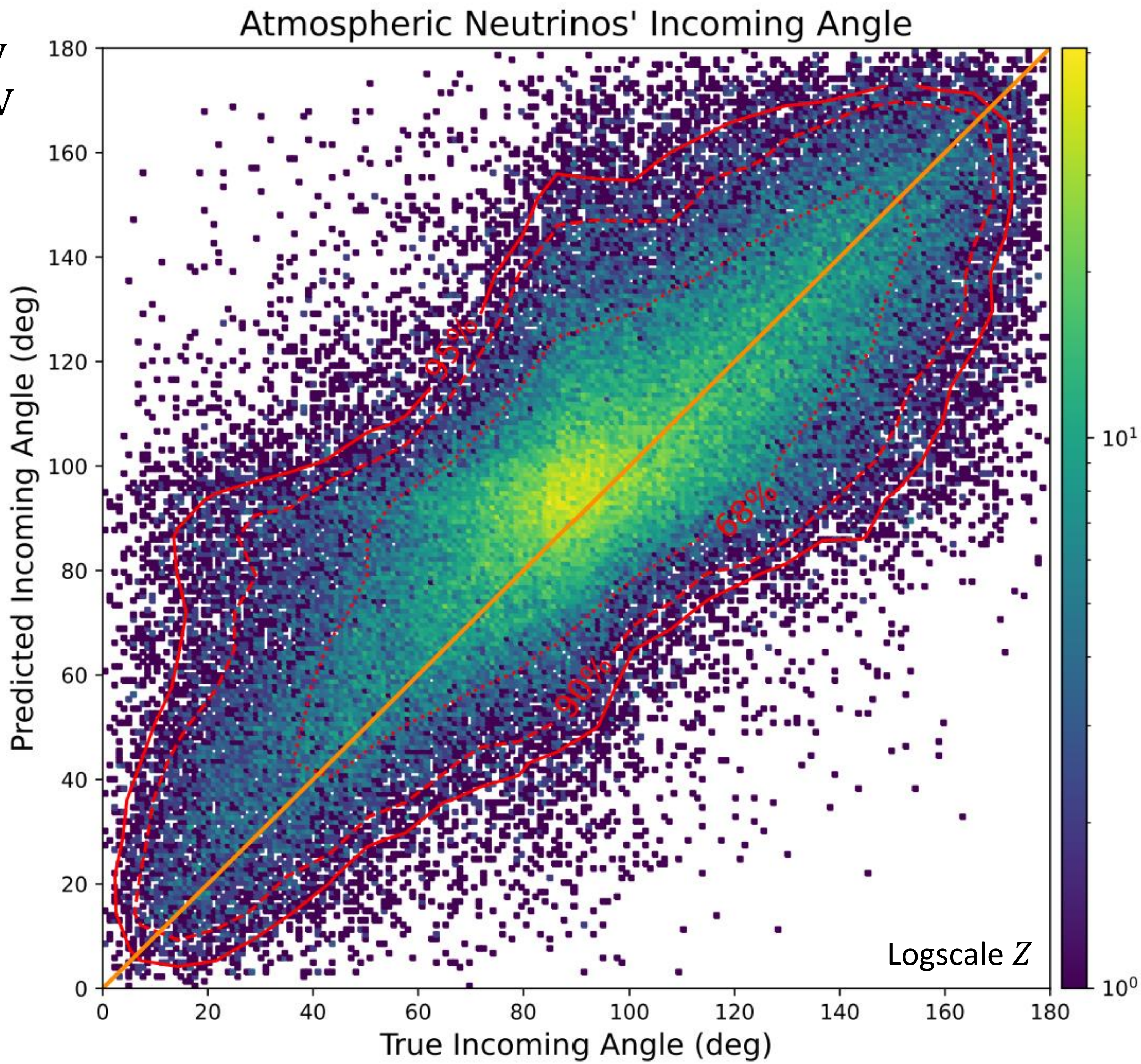
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$

****MAE****

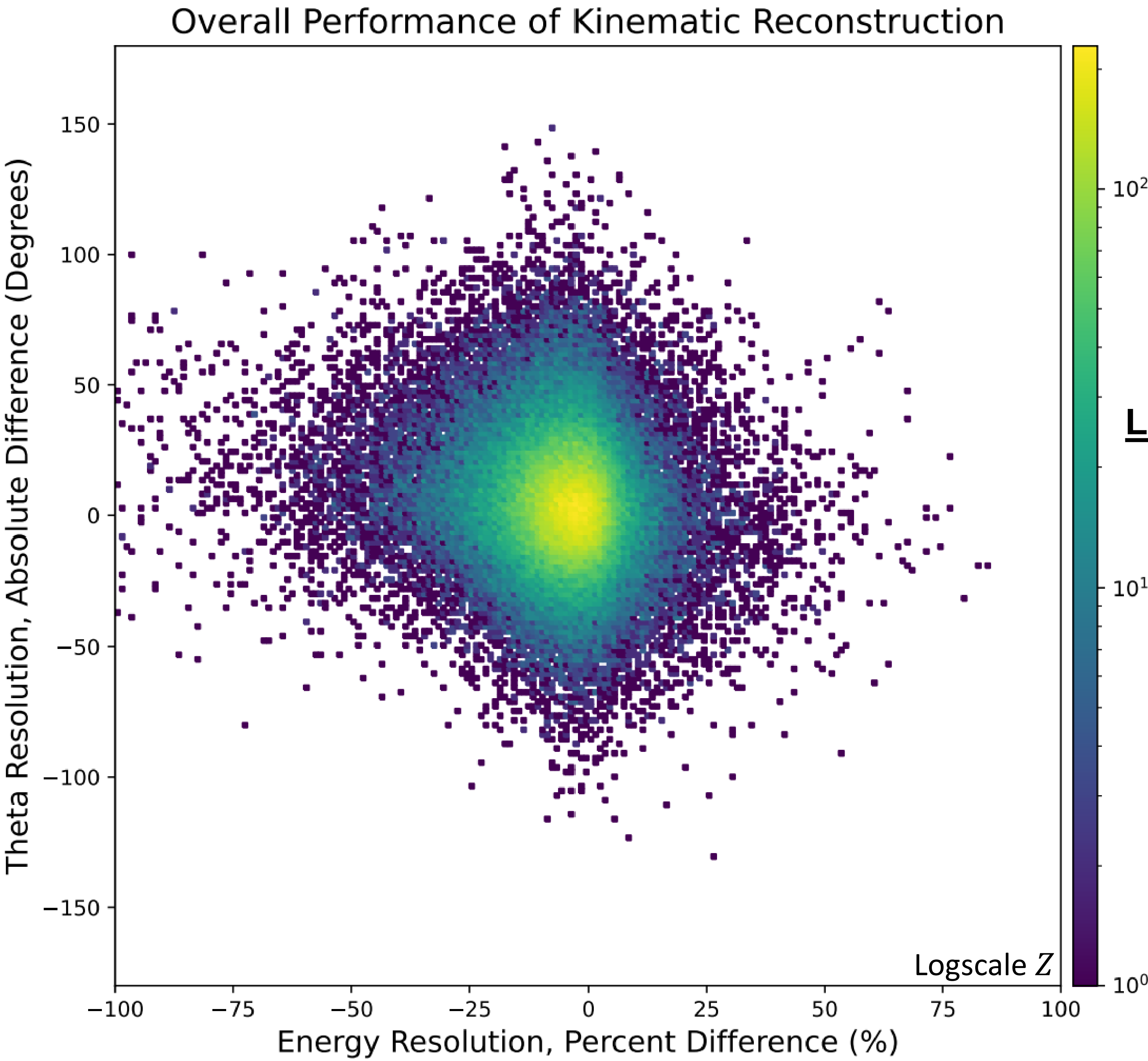
PREVIOUS TRAINING VARIABLES

****CHEATS ON NC EVENTS!!****

```
"vector": [
    "Final_State_Particles_PDG",
    "Final_State_Particles_Mass",
    "Final_State_Particles_Energy",
    "Final_State_Particles_Momentum_X",
    "Final_State_Particles_Momentum_Y",
    "Final_State_Particles_Momentum_Z"

],
"scalar": [
    "Initial_State_Neutrino_PDG",
    "Final_State_Lepton_PDG",
    "Final_State_Lepton_Mass",
    "Final_State_Lepton_Energy",
    "Final_State_Lepton_Momentum_X",
    "Final_State_Lepton_Momentum_Y",
    "Final_State_Lepton_Momentum_Z"

],
"target": [
    "Initial_State_Neutrino_Energy",
    "Initial_State_Neutrino_Momentum_X",
    "Initial_State_Neutrino_Momentum_Y",
    "Initial_State_Neutrino_Momentum_Z"

],
```



Only showing ~100k events here!



Selection:

TRAINING ON $E_\nu \epsilon (0.1,100)\mathrm{GeV}$

SHOWING ONLY $E_\nu \epsilon (0.1,1)\mathrm{GeV}$

****NO THRESHOLDS****

All processes

All ν flavors

All topologies

Any number protons, pions, etc.

NO NEUTRONS

Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

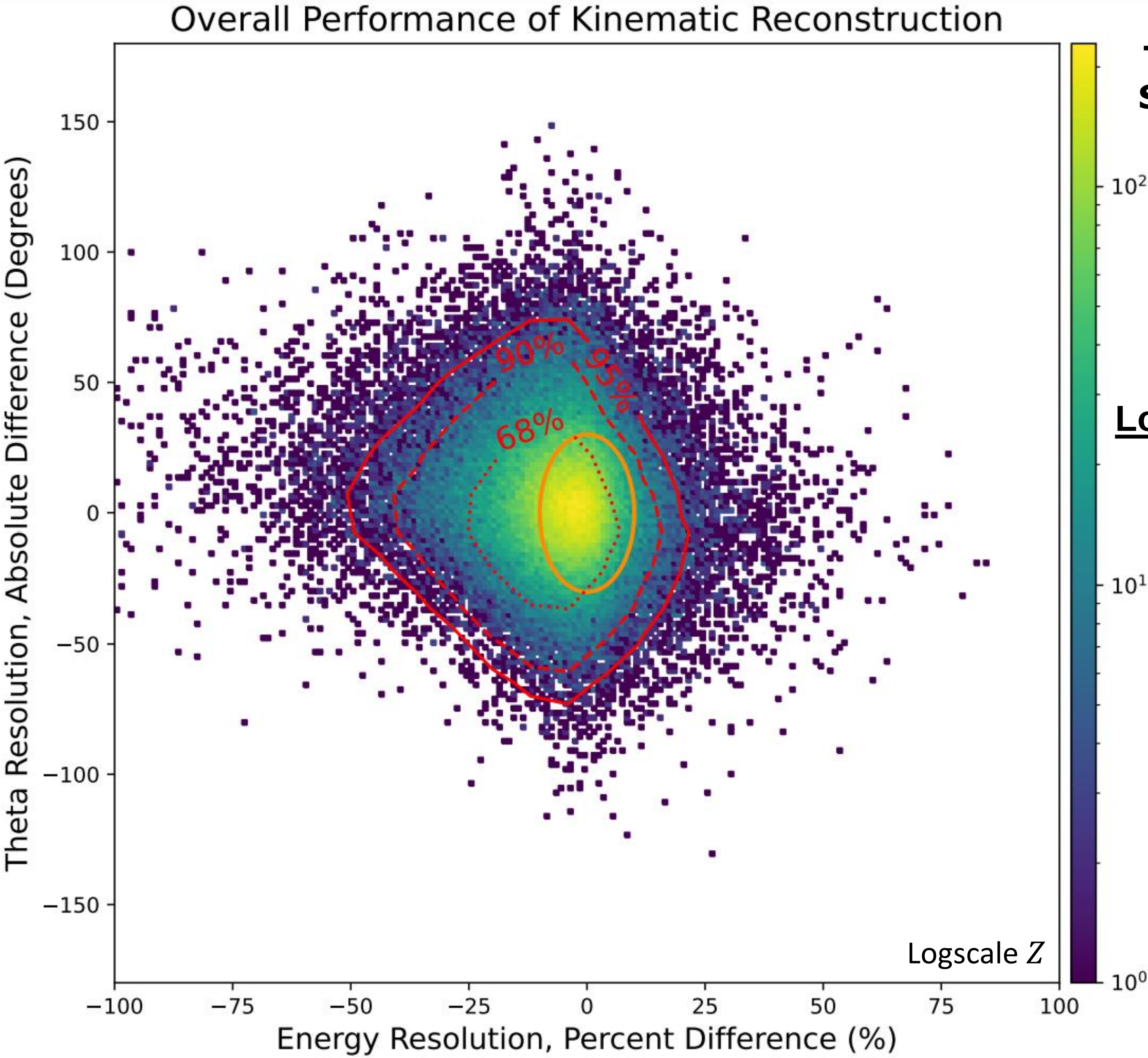
****MAE****

PREVIOUS TRAINING VARIABLES

****CHEATS ON NC EVENTS!!****


```
"vector": [
    "Final_State_Particles_PDG",
    "Final_State_Particles_Mass",
    "Final_State_Particles_Energy",
    "Final_State_Particles_Momentum_X",
    "Final_State_Particles_Momentum_Y",
    "Final_State_Particles_Momentum_Z"

],
"scalar": [
    "Initial_State_Neutrino_PDG",
    "Final_State_Lepton_PDG",
    "Final_State_Lepton_Mass",
    "Final_State_Lepton_Energy",
    "Final_State_Lepton_Momentum_X",
    "Final_State_Lepton_Momentum_Y",
    "Final_State_Lepton_Momentum_Z"

],
"target": [
    "Initial_State_Neutrino_Energy",
    "Initial_State_Neutrino_Momentum_X",
    "Initial_State_Neutrino_Momentum_Y",
    "Initial_State_Neutrino_Momentum_Z"

],
```


Only showing ~100k events here!



Selection:

TRAINING ON $E_\nu \epsilon (0.1,100)\mathrm{GeV}$

SHOWING ONLY $E_\nu \epsilon (0.1,1)\mathrm{GeV}$

****NO THRESHOLDS****

All processes

All ν flavors

All topologies

Any number protons, pions, etc.

NO NEUTRONS

Loss variables: $\left(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu}\right)$

****MAE****

PREVIOUS TRAINING VARIABLES

****CHEATS ON NC EVENTS!!****

```
"vector": [
    "Final_State_Particles_PDG",
    "Final_State_Particles_Mass",
    "Final_State_Particles_Energy",
    "Final_State_Particles_Momentum_X",
    "Final_State_Particles_Momentum_Y",
    "Final_State_Particles_Momentum_Z"

],
"scalar": [
    "Initial_State_Neutrino_PDG",
    "Final_State_Lepton_PDG",
    "Final_State_Lepton_Mass",
    "Final_State_Lepton_Energy",
    "Final_State_Lepton_Momentum_X",
    "Final_State_Lepton_Momentum_Y",
    "Final_State_Lepton_Momentum_Z"

],
"target": [
    "Initial_State_Neutrino_Energy",
    "Initial_State_Neutrino_Momentum_X",
    "Initial_State_Neutrino_Momentum_Y",
    "Initial_State_Neutrino_Momentum_Z"

],
```

Only showing ~100k events here!

Only showing ~100k events here!

Selection:

TRAINING ON $E_\nu \epsilon (0.1, 100)\mathrm{GeV}$

SHOWING ONLY $E_\nu \epsilon (0.1, 1)\mathrm{GeV}$

****NO THRESHOLDS****

All processes

All ν flavors

All topologies

Any number protons, pions, etc.

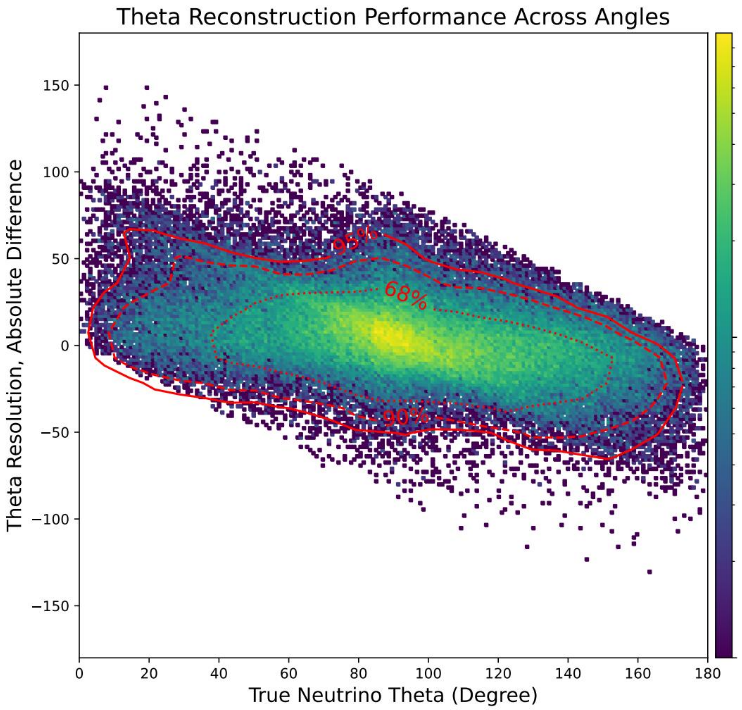
NO NEUTRONS

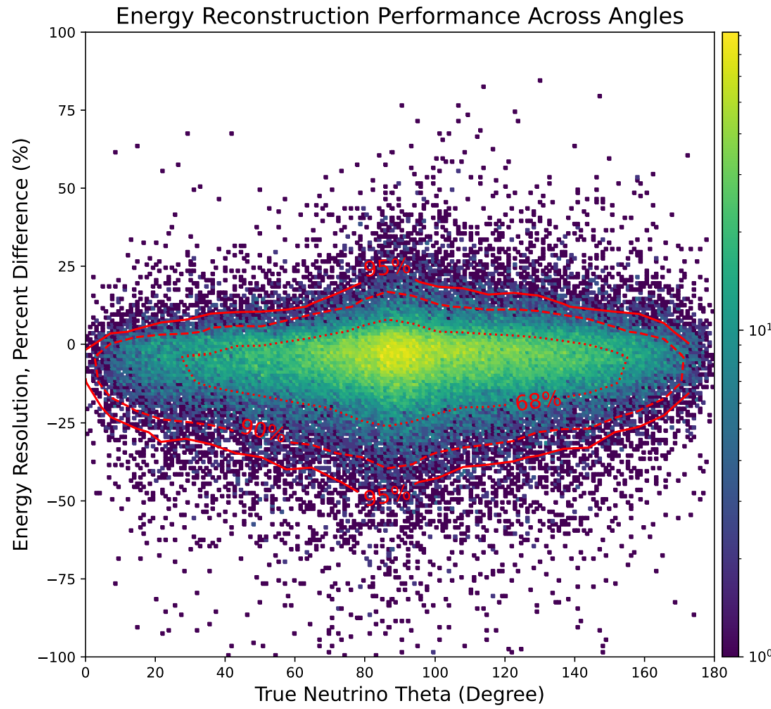
Loss variables: $(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$

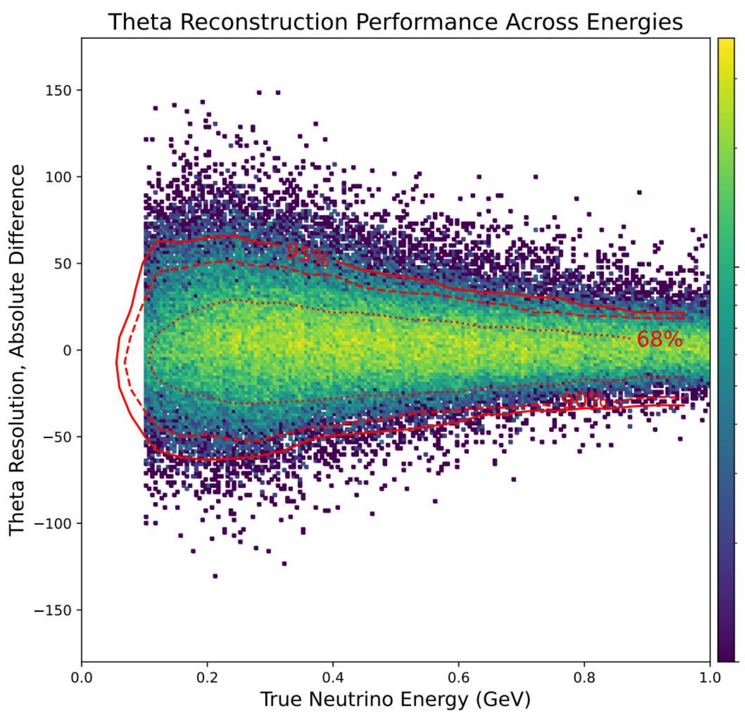
****MAE****

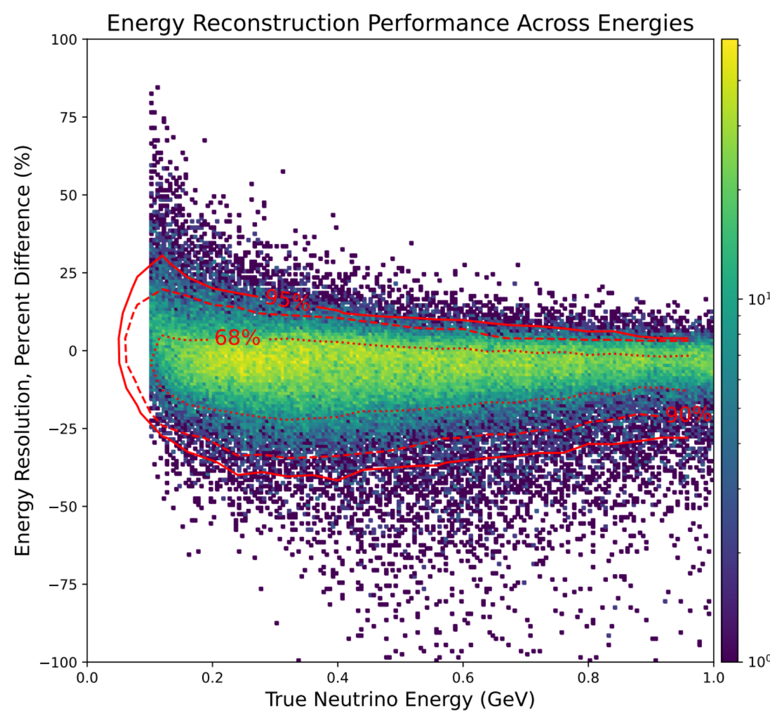
PREVIOUS TRAINING VARIABLES

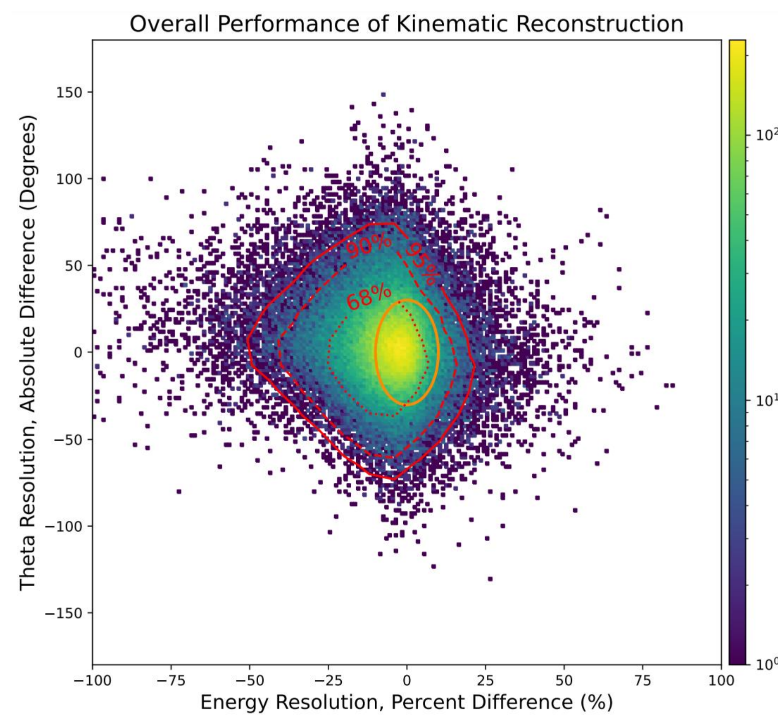
****CHEATS ON NC EVENTS!!****



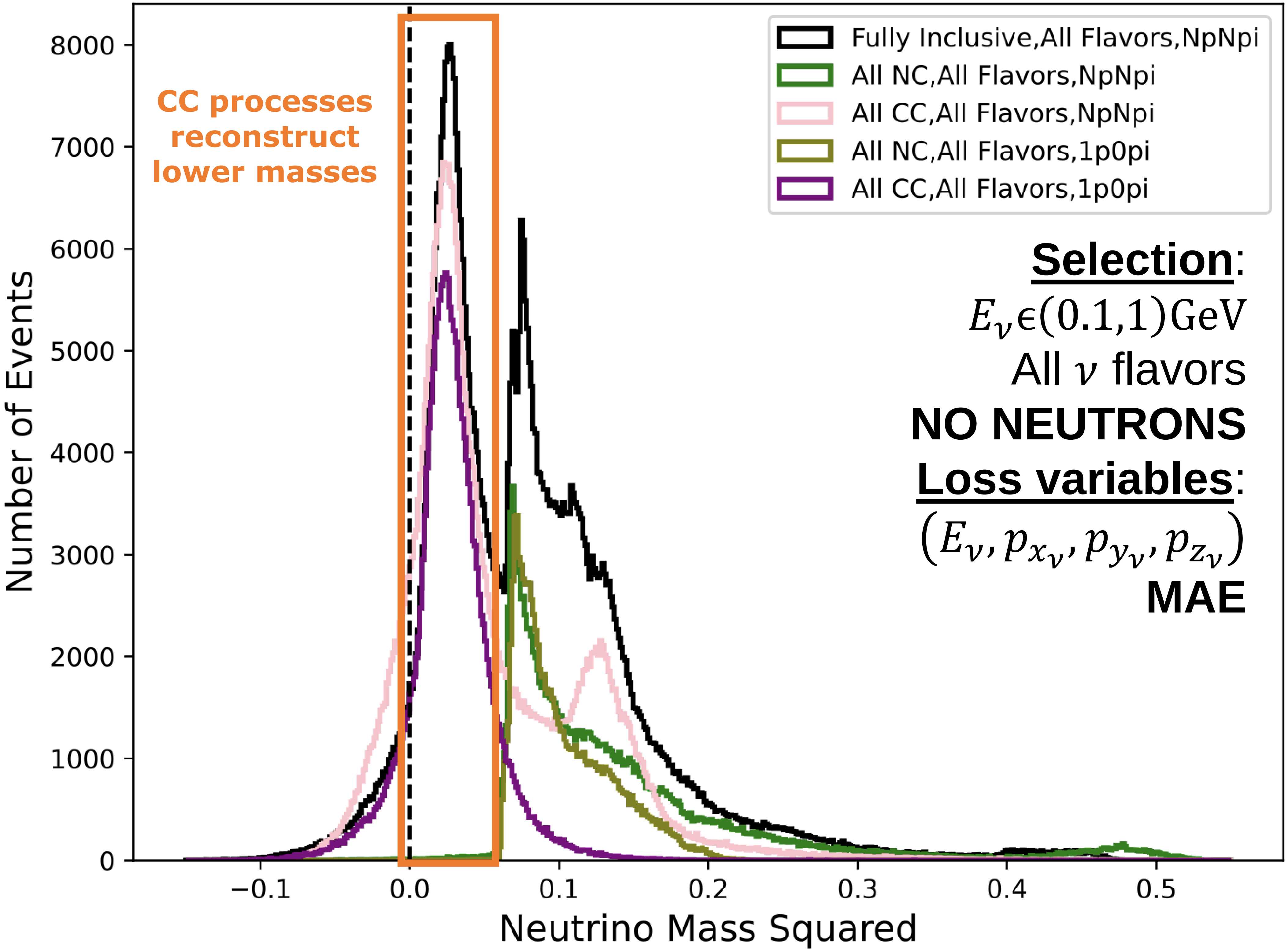




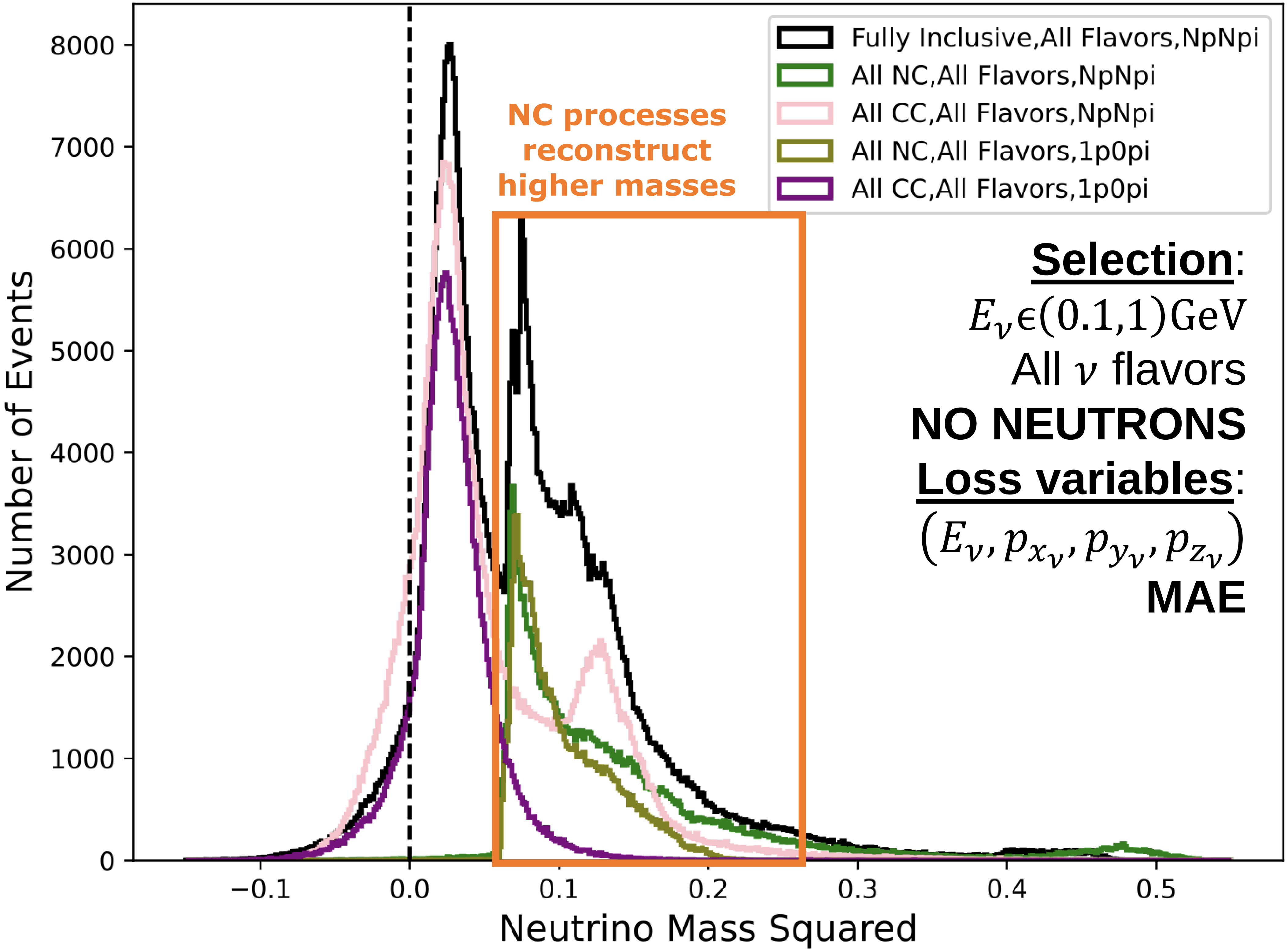




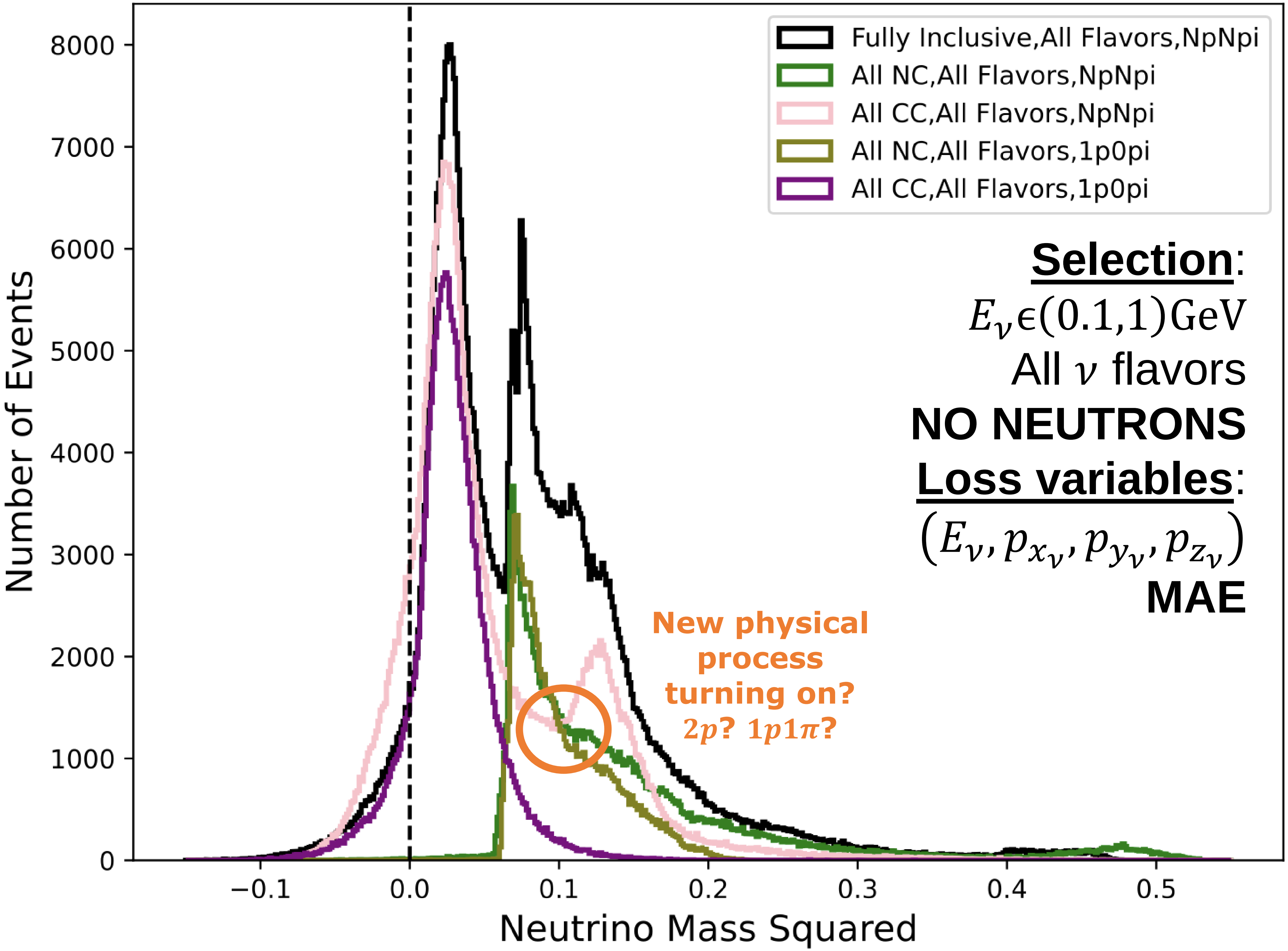
Logscales Z



CC processes reconstruct lower masses
Fully Inclusive,All Flavors,NpNpi
All NC,All Flavors,NpNpi
All CC,All Flavors,NpNpi
All NC,All Flavors,1p0pi
All CC,All Flavors,1p0pi
Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All ν flavors
NO NEUTRONS
Loss variables:
$(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE
Number of Events
Neutrino Mass Squared
0
1000
2000
3000
4000
5000
6000
7000
8000
−0.1
0.0
0.1
0.2
0.3
0.4
0.5



Fully Inclusive,All Flavors,NpNpi
All NC,All Flavors,NpNpi
All CC,All Flavors,NpNpi
All NC,All Flavors,1p0pi
All CC,All Flavors,1p0pi
NC processes reconstruct higher masses
Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All ν flavors
NO NEUTRONS
Loss variables:
$(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE
Number of Events
Neutrino Mass Squared
0
1000
2000
3000
4000
5000
6000
7000
8000
−0.1
0.0
0.1
0.2
0.3
0.4
0.5



Fully Inclusive,All Flavors,NpNpi
All NC,All Flavors,NpNpi
All CC,All Flavors,NpNpi
All NC,All Flavors,1p0pi
All CC,All Flavors,1p0pi
Selection:
$E_\nu \epsilon (0.1,1)\mathrm{GeV}$
All ν flavors
NO NEUTRONS
Loss variables:
$(E_\nu, p_{x_\nu}, p_{y_\nu}, p_{z_\nu})$
MAE
New physical process turning on?
$2p$? $1p1\pi$?
Number of Events
Neutrino Mass Squared
0
1000
2000
3000
4000
5000
6000
7000
8000
−0.1
0.0
0.1
0.2
0.3
0.4
0.5

Summary & Discussion

- Initial forays: GENIE-only simultaneous energy/angle prediction promising
  - Need to study more effects of…
    - Topological selection, CC & NC processes, nuclear model configuration, training sizes, loss function styles, target/predicted variables, etc…
  - New GENIE samples ready to go, will be running over many of them…
- **Will be considering flattening fluxes**
  - Atmospherics have very hard spectral shape—want to make tool broadly useful
- **Plan**:
  - Test including lepton scalar features once again, for training on at least CC-only
  - Consider different loss functions and loss variable combinations…
    - Angle directly, P_{miss}, baseline directly, others…we have completed many of these already!
    - LSTM style with ν and ℓ playing a role in loss, combinations thereof—p_{ν}^{μ} & $p_{\ell}^{\mu} \Longrightarrow Q^2$?
    - Loss function of p_{ν}^{μ} directly—minimize to make ν mass as small as possible!
    - Consider different loss function coefficients—may help improve angular resolution?
  - Come out with *GENIE-only* "concept" paper: LE, ME, HE atmospherics *and* beam
  - Target two reconstruction papers in DUNE and NOvA separately
    - DUNE: atmospherics in FD, beam in ND?—new atmospheric productions ready for this
    - NOvA: beam events in ND—target improved cross section measurements

Thank-you for your attention!

Questions?

Comments?

Backups

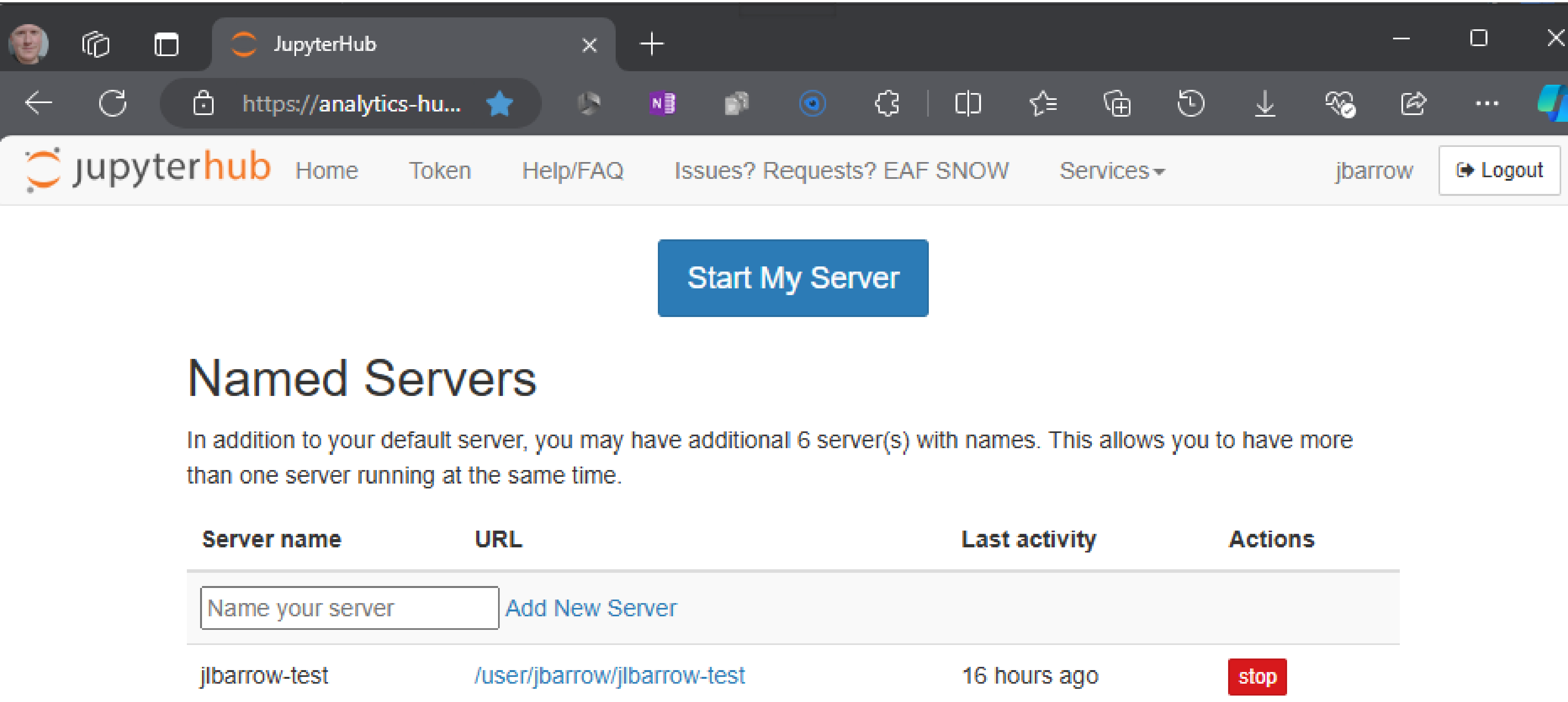
Data Frames Loaded via .csv Files

Event_Index	Initial_State_Neutrino_PDG	Initial_State_Neutrino_Energy	Initial_State_Neutrino_Momentum_X	Initial_State_Neutrino_Momentum_Y	Initial_State_Neutrino_Momentum_Z	Initial_Neutrino_CosTheta
0	12	1.48	-0.268	0.747	-1.24	0.506
1	12	0.901	0.124	-0.246	-0.857	-0.273
2	-14	1.92	0.459	1.37	-1.27	0.712
3	12	1.91	-1.02	-0.188	-1.61	-0.0985
4	14	0.308	-0.258	0.0248	-0.167	0.0805
5	-14	0.264	0.255	0.0118	0.067	0.0446
6	12	1.02	0.907	0.0737	0.458	0.0723
7	14	8.35	0.251	5.38	-6.39	0.644
8	16	3.71	2.97	-0.413	-2.18	-0.111
9	-16	5.62	-5.34	1.23	1.24	0.219
10	14	3.23	2.24	-2.05	1.08	-0.636
11	14	0.904	0.123	-0.683	0.58	-0.755
12	12	3.27	2.41	2.17	-0.404	0.663
13	14	16.7	-2.52	15.9	-4.28	0.955
14	14	0.425	-0.413	-0.083	-0.0529	-0.196
15	12	2.29	0.276	1.2	-1.93	0.526
16	14	1.13	0.276	0.623	-0.898	0.553
17	12	1.76	-1.74	0.244	-0.0585	0.138
18	14	10.5	-3.47	8.74	4.67	0.833
19	14	0.216	-0.156	-0.0197	-0.148	-0.0913
20	12	3.31	-1.47	0.0594	-2.97	0.0179
21	-12	1.32	-1.09	0.23	0.71	0.174
22	12	0.555	-0.547	-0.0653	-0.0645	-0.118
23	12	0.384	0.148	-0.319	0.155	-0.829
24	14	0.725	0.221	0.69	-0.00953	0.952
25	12	0.552	0.372	0.291	-0.286	0.527
26	14	0.605	-0.487	0.179	-0.311	0.296
27	14	0.156	-0.0498	-0.143	0.0387	-0.915
28	12	0.36	-0.0712	-0.348	0.0595	-0.966
29	-12	4.27	-2.09	-0.261	3.72	-0.0612
30	12	0.182	-0.122	-0.0628	-0.119	-0.345
31	14	1.54	0.42	0.258	1.46	0.167
32	14	0.198	0.0311	0.00666	-0.195	0.0337

S. Wu

- Polars replacing Pandas for data handling possible
  - Development branch: [wswxyq/transformer_EE at polars]
  - Smarter than Pandas, more memory efficient for CSV loading

[FNAL Elastic Analysis Facility]
[EAF Documentation]



[FNAL Elastic Analysis Facility](#)
[EAF Documentation](#)

Server Options

GPUS (used/capacity):
10GB (1/20), **20GB** (13/20), **40GB** (4/20)

CMS
CVMFS, HTCondor, COFFEA

- CPU Interactives: AL9
- NVIDIA® A100 GPU: AL9 - 10GB GPU slot

LBNF DUNE/ProtoDUNE
CVMFS, LarSoft

- CPU Interactives: AL9
- NVIDIA® A100 GPU: AL9 - 20GB GPU slot
  - AL9 - 20GB GPU slot
  - AL9 - 40GB GPU slot
  - AL9 - 10GB GPU slot
  - SL7 - 20GB GPU slot
  - SL7 - 40GB GPU slot
  - SL7 - 10GB GPU slot

FIFE
CVMFS Neutrinos/Mu2e/gm2

- CPU Interactives: AL9
- NVIDIA® A100 GPU: AL9 - 10GB GPU slot

Astro/Cosmic Frontier
CVMFS, LSST kernel

- CPU Interactives: AL9 (LSST kernel)

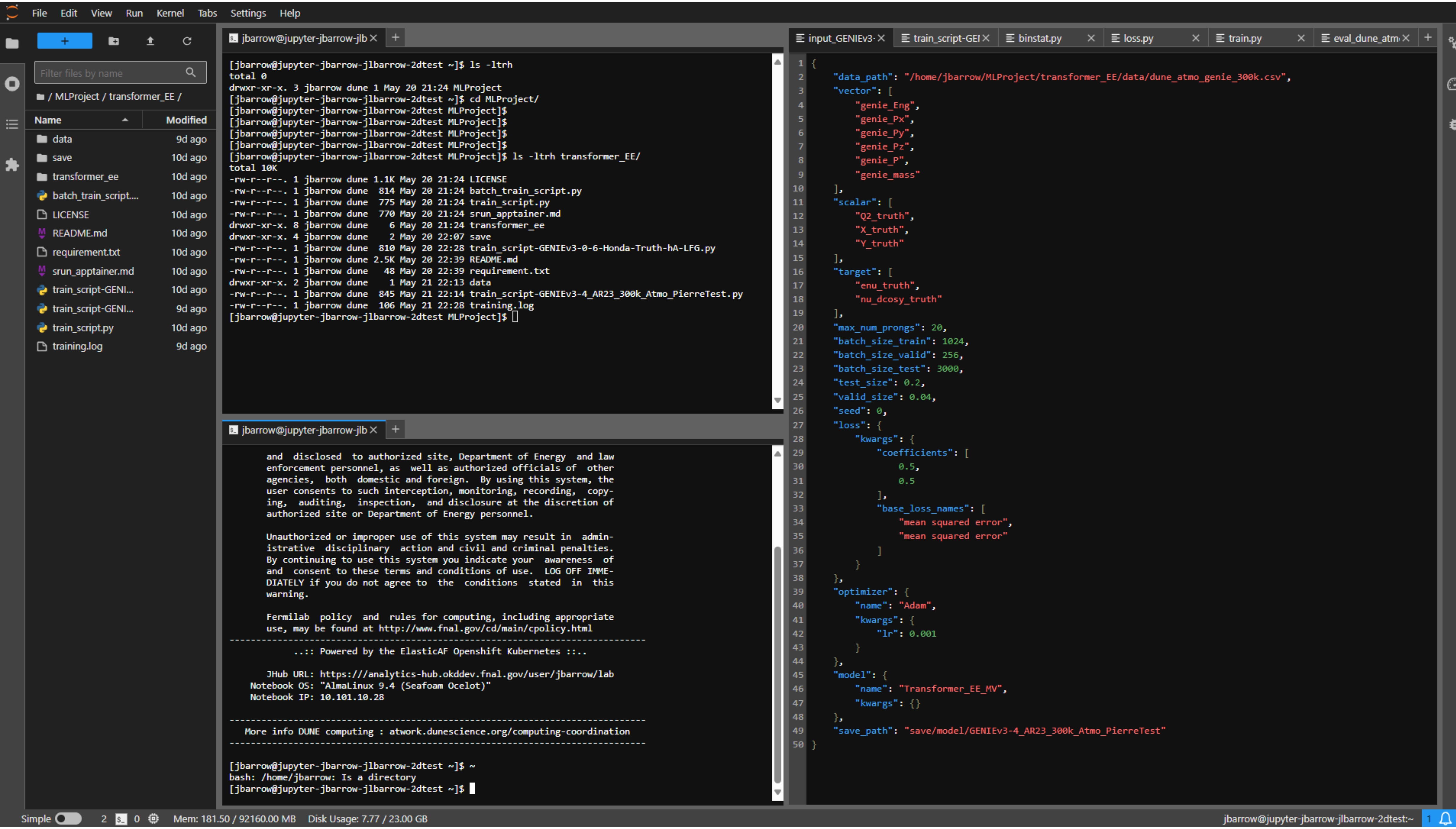
ACCEL-AI
Tensorflow, pyTorch

- CPU Interactives: SL7 L-CAPE

ACORN
ACSYS python, Fortran

- CPU Interactives: AL9

[FNAL Elastic Analysis Facility](#)
[EAF Documentation](#)



```python
def linear_combination_loss(output, target, weight=None, **kwargs):
    """
    linear combination of base loss functions
    coefficients, base_loss_names should have the same length, which is the number of output variables
    e.g. kwargs = {"coefficients": [0.5, 0.5], "base_loss_names": ["mean squared error", "mean absolute error"]}
    """
    if "base_loss_names" not in kwargs or "coefficients" not in kwargs:
        raise ValueError("base_loss_names and coefficients must be provided in kwargs")

    if len(kwargs["base_loss_names"]) != len(kwargs["coefficients"]):
        raise ValueError(
            "base_loss_names and coefficients must have the same length\n",
            "len(base_loss_names):",
            len(kwargs["base_loss_names"]),
            "\nlen(coefficients):",
            len(kwargs["coefficients"]),
        )

    base_loss_names = kwargs["base_loss_names"]
    coefficients = kwargs["coefficients"]
    linear_loss = 0
    for i in range(len(base_loss_names)):
        linear_loss += coefficients[i] * loss_function[base_loss_names[i]](
            output[:, i], target[:, i], torch.squeeze(weight)
        )
    return linear_loss
```

```python
import json

from transformer_ee.train import MVtrainer

#with open("transformer_ee/config/input_DUNE_atmo.json", encoding="UTF-8", mode="r") as f:
#with open("transformer_ee/config/input_DUNE_atmo-4m.json", encoding="UTF-8", mode="r") as f:
with open("/home/jbarrow/MLProject2/transformer_EE/transformer_ee/config/input_GENIEv3-0-6-Honda-Truth-hA-LFG_wLeptonScalars.json", encoding="UTF-8", mode="r") as f:
    input_d = json.load(f)


input_d["data_path"]="/exp/dune/app/users/jbarrow/MLProject/AtmoNu_hA_BR_wAngles_1M.csv"
# input_d["model"]["name"] = "Transformer_EE_v4"
input_d["model"]["kwargs"]["nhead"] = 2
input_d["model"]["epochs"] = 100
input_d["model"]["kwargs"]["num_layers"] = 5
#input_d["optimizer"]["name"] = "sgd"
input_d["optimizer"]["name"] = "Adam"
input_d["optimizer"]["kwargs"]["lr"] = 0.001
#input_d["optimizer"]["kwargs"]["momentum"] = 0.9
input_d["save_path"] = "/home/jbarrow/MLProject2/save/model/GENIEv3-0-6-Honda-Truth-hA-LFG_wLeptonScalars_MAE/"
# input_d["weight"] = {"name": "FlatSpectraWeights", "kwargs": {"maxweight": 5, "minweight": 0.2}}


my_trainer = MVtrainer(input_d)
my_trainer.train()
my_trainer.eval()
```


```json
{
    "data_path": "/home/jbarrow/MLProject/transformer_EE/data/dune_atmo_genie_300k.csv",
    "vector": [
        "genie_Eng",
        "genie_Px",
        "genie_Py",
        "genie_Pz",
        "genie_P",
        "genie_mass"
    ],
    "scalar": [
        "Q2_truth",
        "X_truth",
        "Y_truth"
    ],
    "target": [
        "enu_truth",
        "nu_dcosy_truth"
    ],
    "max_num_prongs": 20,
    "batch_size_train": 1024,
    "batch_size_valid": 256,
    "batch_size_test": 3000,
    "test_size": 0.2,
    "valid_size": 0.04,
    "seed": 0,
    "loss": {
        "kwargs": {
            "coefficients": [
                0.5,
                0.5
            ],
            "base_loss_names": [
                "mean squared error",
                "mean squared error"
            ]
        }
    },
    "optimizer": {
        "name": "Adam",
        "kwargs": {
            "lr": 0.001
        }
    },
    "model": {
        "name": "Transformer_EE_MV",
        "kwargs": {}
    },
    "save_path": "save/model/GENIEv3-4_AR23_300k_Atmo_PierreTest"
}
```

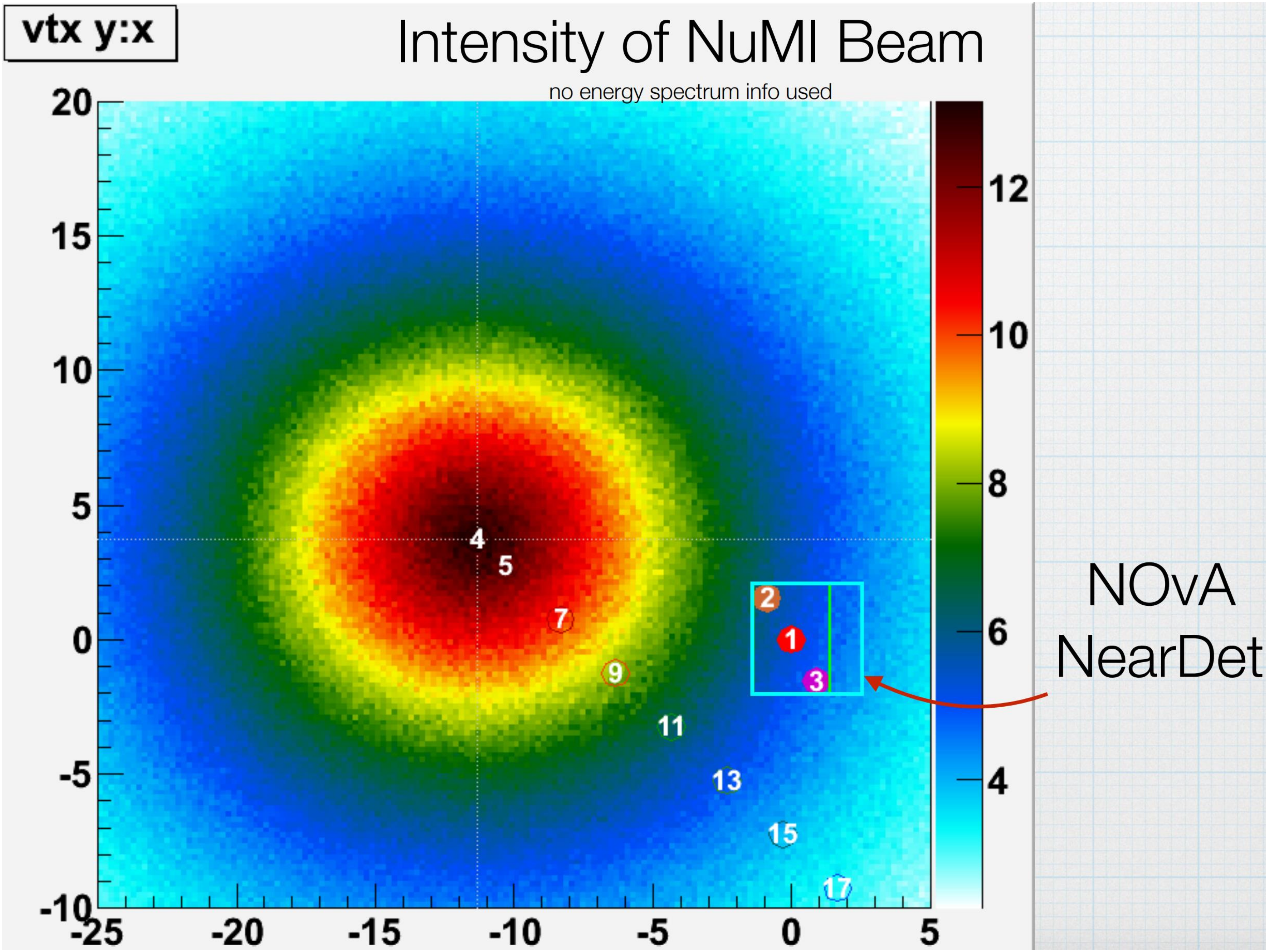

Input .json file

Configures training variables

Configures target variables

- Used in composite loss

Configures individual loss functions per variable



[Robert Hatcher: Preliminary schema for a common flux/geometry driver]

How to Potentially Improve Energy Estimators?

- Proposing some basics for update to LSTM_EE:
  - **Include angular factor in loss function**
    - **Should keep training until some kinematic consistency found**
      - Hopefully will improve energy estimation given extra kinematic constraint
    - Should be **aware of *true* (p_x, p_y, p_z) *and predicted* $\left(p_x, p_y, p_z\right)$**
      - Many variable already included as input, but not predicted as output
      - Currently only the energy enters the loss function
      - Some features could already be "subliminally" informing angular reco…

  - **Include buffer between true & reco. angle in loss**
    - Need to make sure not to overtrain given…
      - Nuclear modeling biases
      - Neutrons, nuclear remnants
        - Plan to include input/output with/without neutrons/HadrBlobs
      - Detector resolution/reconstruction issues
        - Prongs in NOνA have limiting resolution…

  - Study effects of inclusive/exclusive CC training samples
    - Will be topologically based on prong multiplicities (most likely)
    - Does energy resolution improve with angular constraint?
    - Does energy resolution improve with specific kinds of topologies?

  - Should loss function be non-linear?

SAMPLES (ATMONU AND NU BEAM)		
Signal Selection	**True nu Energy Range (GeV)**	**File Location**
Inclusive (CC,NC)	0.1 - 1.0	
Inclusive (CC,NC)	0.1 - 5.0	
Inclusive (CC,NC)	0.1-10.0	
numuCC1p0pi	0.1-1.0	
numuCC1p0pi	0.1-5.0	
numuCCNp0pi	0.1-5.0	
numuCCNp0pi	0.1-10.0	
nueCC1p0pi	0.1-1.0	
nueCC1p0pi	0.1-5.0	
nueCCNp0pi	0.1-5.0	
nueCCNp0pi	0.1-10.0	
numuCCX	0.1-1.0	
numuCCX	0.1-5.0	
numuCCX	0.1-10.0	
nueCCX	0.1-1.0	
nueCCX	0.1-5.0	
nueCCX	0.1-10.0	
NC1p0pi	0.1-1.0	
NC1p0pi	0.1-5.0	
NCNp0pi	0.1-10.0	

- Currently considering truth-only samples
  - GENIEv3.0.6
  - Uses fully oscillated Honda flux
    - Homestake site
      - $15\mathrm{km}$ production height
    - Hack to put in $\nu_\tau, \bar{\nu}_\tau$
- May need to remake all of this with flattened fluxes
  - Fast falloff of spectral index limits training on high energy events
    - $\sim E_\nu^{-2.5}$
- Beam events “ready to go”
  - Currently have NuMI flux files
  - Again, may need to flatten

- Signal selection to check performance
  - $\mathrm{CC1p0}\pi$ focus of Pedro *et al*’s paper

- Energy ranges to study validity
  - $0.1 - 1.0\mathrm{GeV}$ *a la* Pedro *et al*
  - $0.1 - 5.0\mathrm{GeV}$
    - $0.1 - 4.0$ studied by Farrell and Higuera for their CVN tool to identify nueCC, numuCC, NC
    - $\sim 5.0\mathrm{GeV}$ is containment cutoff in reduced geometry
  - $0.1 - 10.0\mathrm{GeV}$
    - Rough expected containment in full geometry

COMPOSITE LOSS FUNCTION IDEAS					
All Kinematic Variables (including training)	**Loss Variable Combinations**	**Loss Function Type**	**Proposed Form**	**Best Model Location**	**Model Results Location**
E_nu	(E_nu,px_nu,py_nu,pz_nu)	MSE?			
px_nu,py_nu,pz_nu	(E_nu,px_nu,py_nu,pz_nu,P_miss)	MSE????			
E_l,px_l,py_l,pz_l,p_l,KE_l	(E_nu,px_nu,py_nu,pz_nu,baseline)	MSE			
E_p,px_p,py_p,pz_p,p_p,KE_p	(E_nu,baseline)	MSE			
theta_nu, costheta_nu, phi_nu	(E_nu,theta_nu)	MSE			
theta_l, costheta_l,phi_l	(E_nu,theta_nu,baseline)	MSE			
baseline	(E_nu,costheta_nu)	MSE			
P_miss	(E_nu,costheta_nu,baseline)	MSE			
KE_tot	(E_nu,theta_nu,phi_nu)	MSE			
	(E_nu,costheta_nu,phi_nu)	MSE			

LARTPC-LIKE PARTICLE TRACKING THRESHOLDS	
Particle Type	**KE Minimum (MeV)**
Proton	25?
Pi+-	70?
Pi0	50?
K+-	50?
Muon	5
Electron	5

Want to be able to easily configure these…All code developed!

Want to be conservative! But made with LArTPCs in mind…

Will update when moving to NOvA-oriented analysis
→NOvA reconstructed prongs, etc…