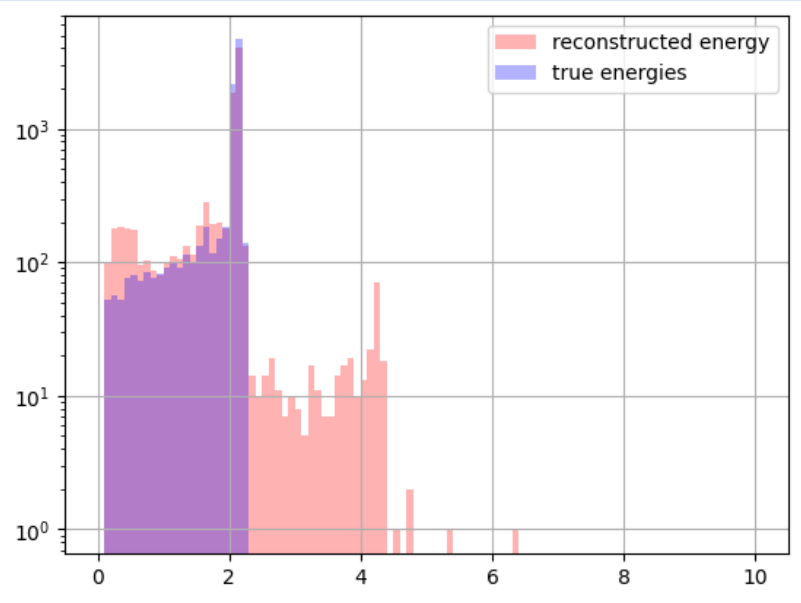
Notes

- Select 3 events and try to get 0 loss (maybe with lr = 5e-3) → **started, get almost 0 loss (we get not 0 loss, maybe hint that using only 8 features is not correct model)**
- Use 12 features (8 + 4 delta features) → try to increase number of nodes. Does this improve loss?
- Create validation data → **done**
- Test which of the Δfeatures is the relevant one
- Add also mean features: E1+E1, theta1+theta2,ph1+phi2,time1+time2

  → does adding up angles make really sense? I doubt...
- Cleanup of code → almost done
- Plot Energy reconstruction with “Edge method”, classic califa clustering, agglomerative clustering, → done

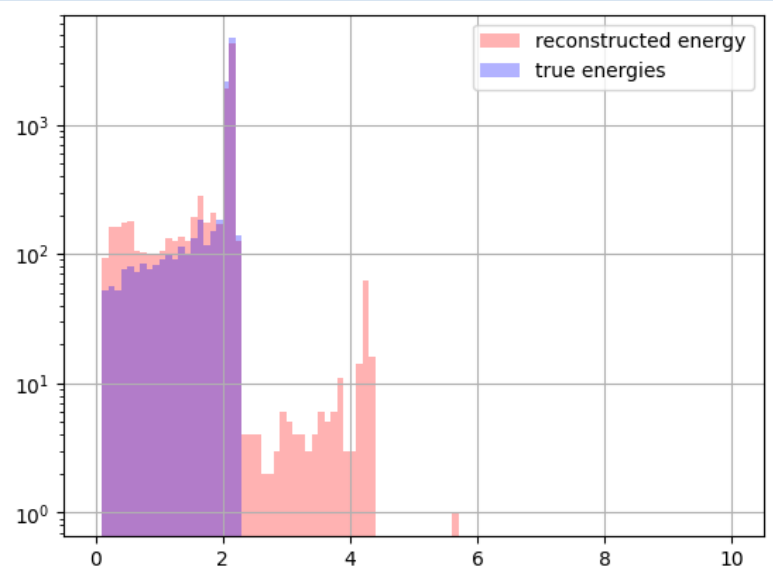
Comparison Clustering Models?

Standard R3B Clustering



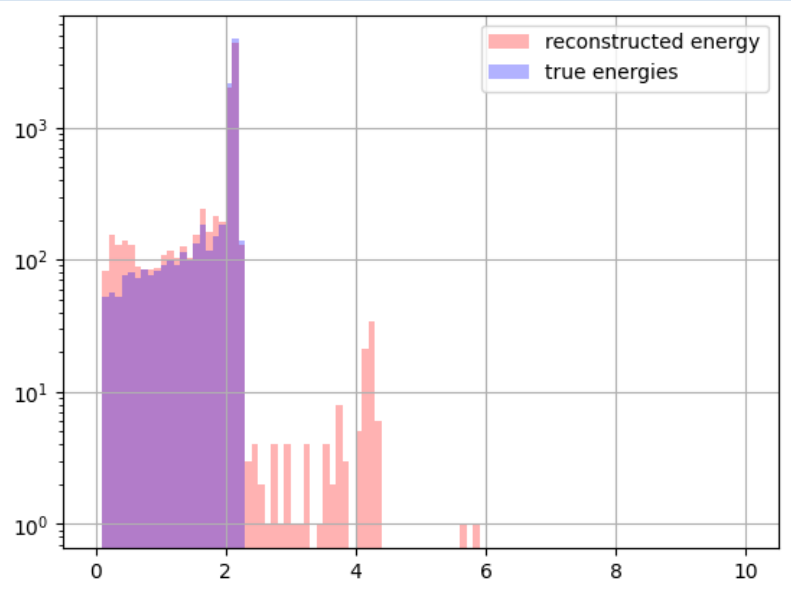
Well reco: 86.9 %

Agglomerative Clustering



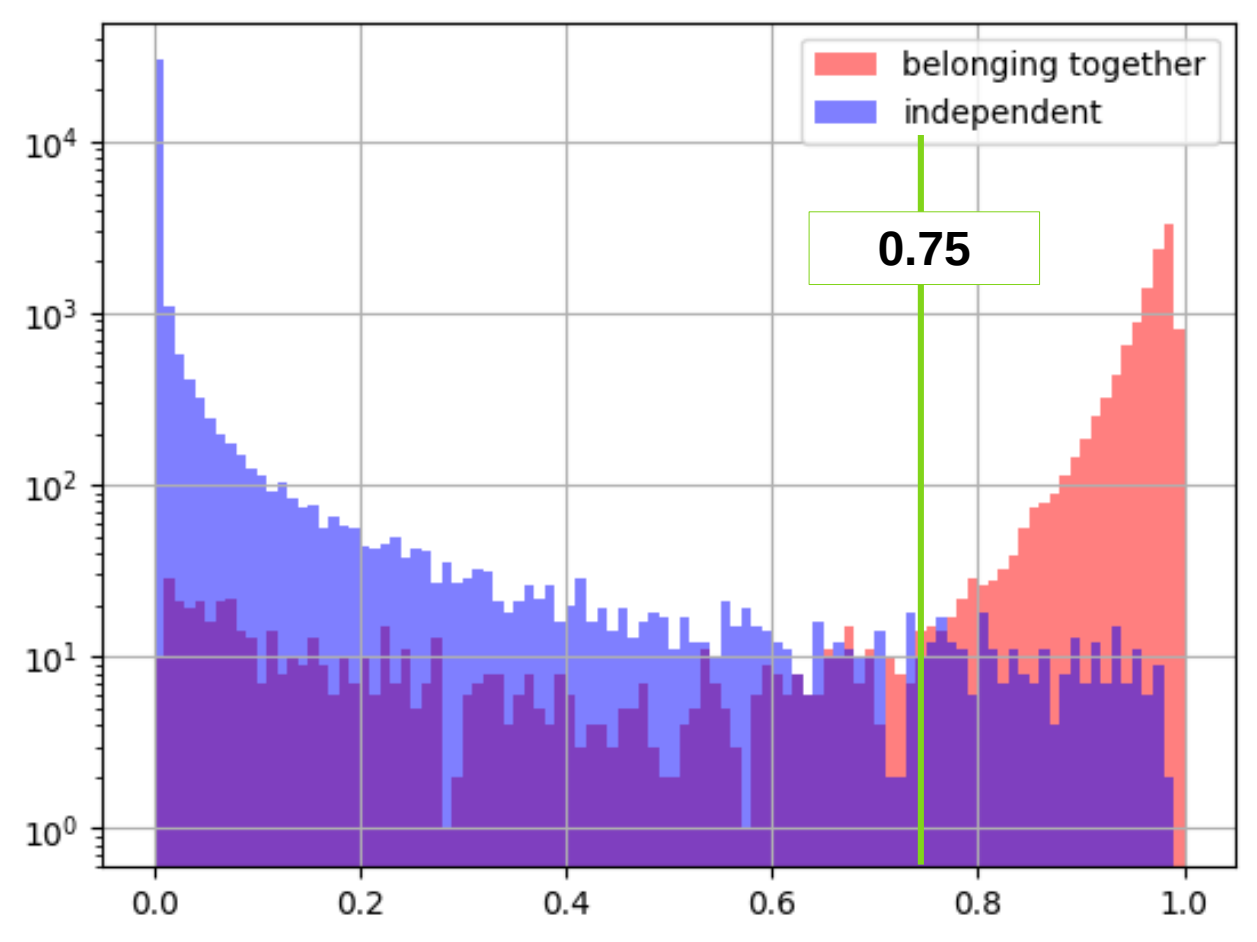
Well reco: 89.9 %

“Edge” Clustering



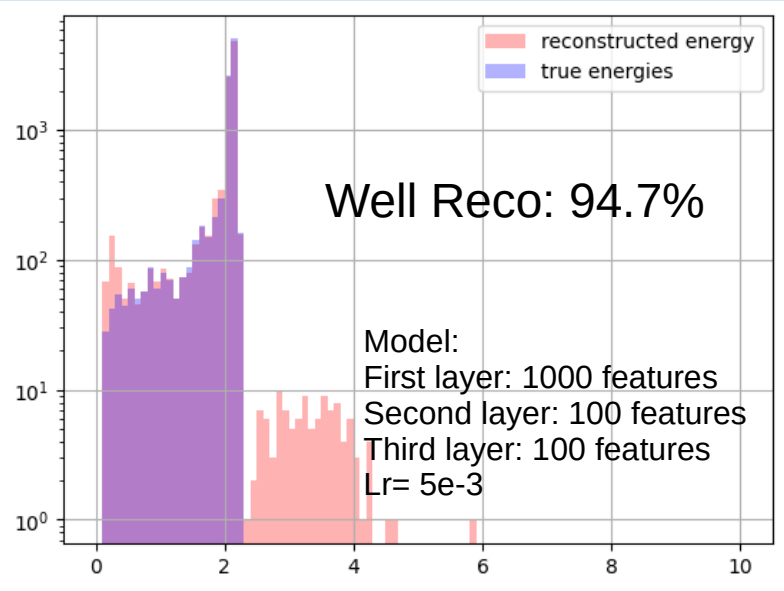
Well reco: 93.4 %

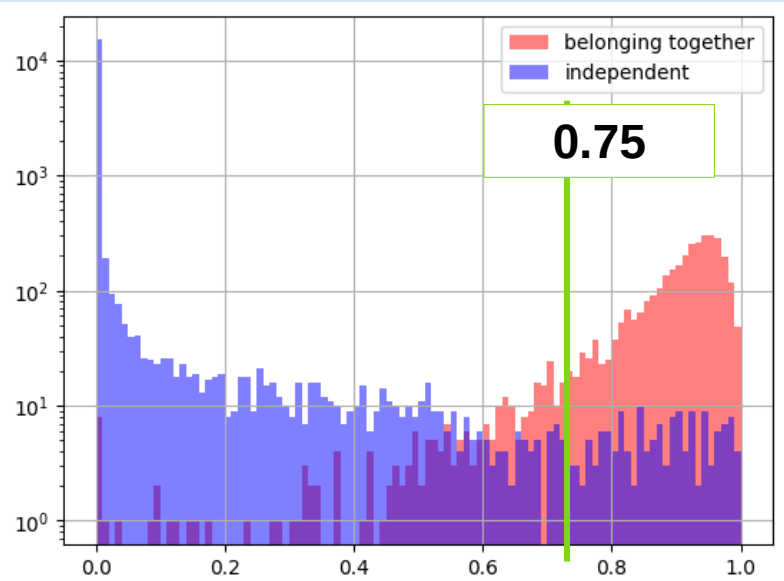
Prediction Values for “Edge” Clustering Model

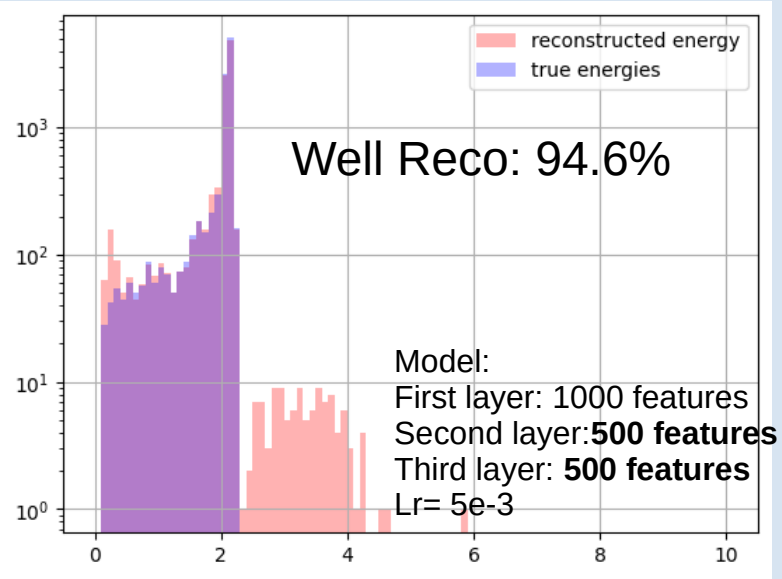


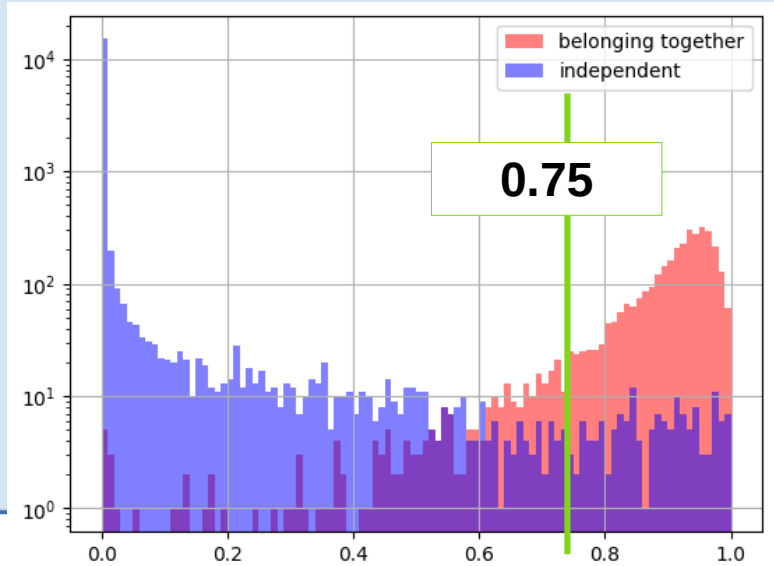
Model:
First layer: 1000 features
Second layer: 100 features
Third layer: 100 features
Lr= 5e-3

Back to False Negative Data









What else?

- Validation/Training data: now available for full dataset and for “false negative events”
- Code was cleaned up with coherent procedures/methods for all three clustering methods → helpful in future, less error prone
- Test to bring loss value to 0 (using only 3 events) → only tested using the 8 features is not equal 0 event after 10e5 iterations, however it is monotonically decreasing high lr (~$10^{-}1$) are better

TODOS:

- Add also mean features: E1+E1, theta1+theta2,ph1+phi2,time1+time2
- Run “Edge Model” on validation data. Do I still get a reco rate of 93%?
- Build “Split Model” to predict if preclustered data should be split apart or not
- What about give transformer now a try with Δfeatures as input data?

Other?