# GNN fit robustness II

06.09.24

# “Mixing”

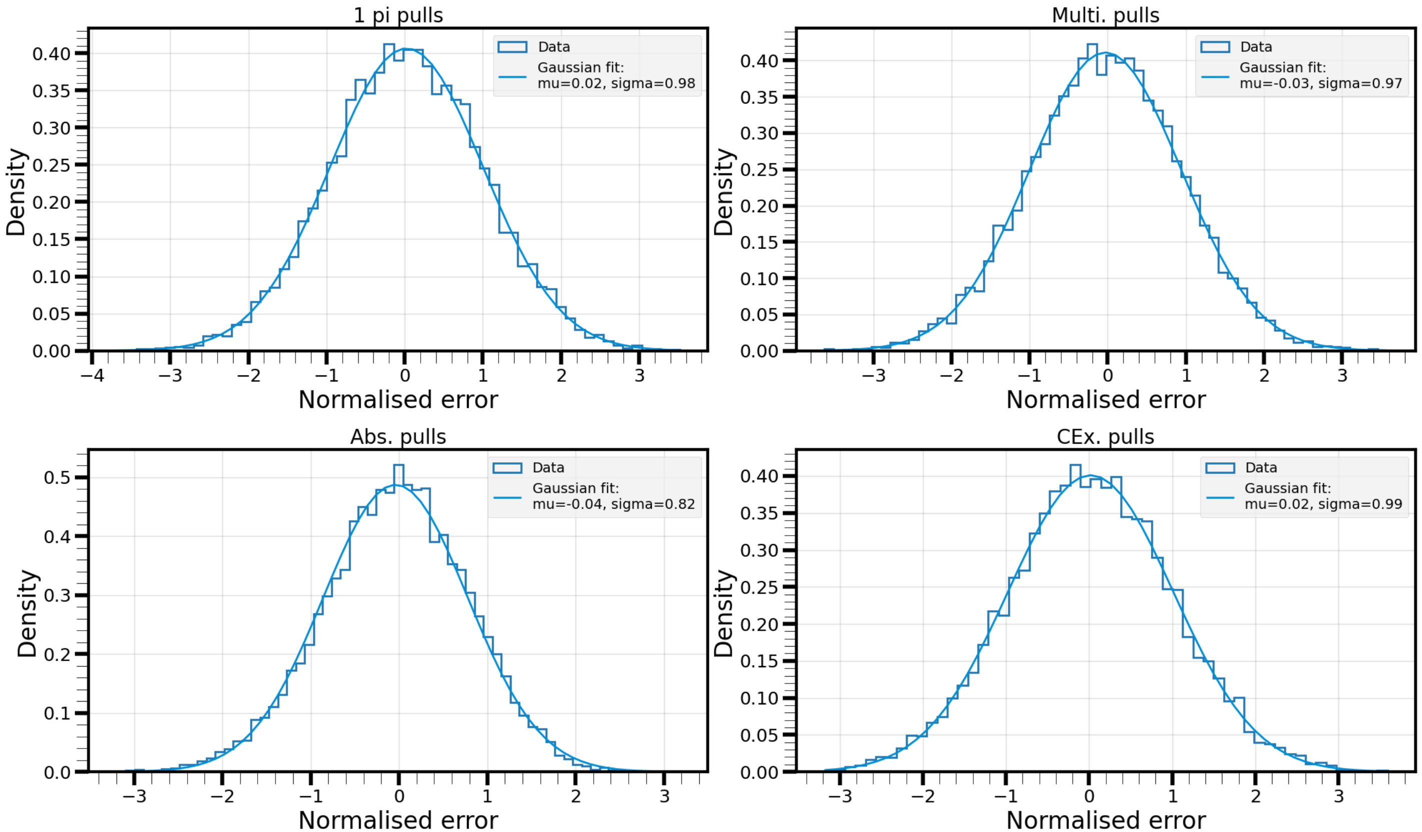
- Same source for all MC – draw from “true” distribution
- Previously, this was divided into two samples (files).
- From these, sub-samples were created.
- Treating the “true” distribution as a multinomially distributed set of bins.
- When sampling from the subsample, we sample with a probability $\frac{X}{N_{sub}} = p_{i,sub} \neq p_{i,true}$, where $X \sim \mathrm{Multi}(N_{full}, p_{i,true})$ from the true multinomial.

# “Mixing”

- Before showing any results:
- How large a pull is acceptable?
  - I’ve already looked at a bunch of results, so don’t trust myself to think about it sensibly

# “Mixing” pulls

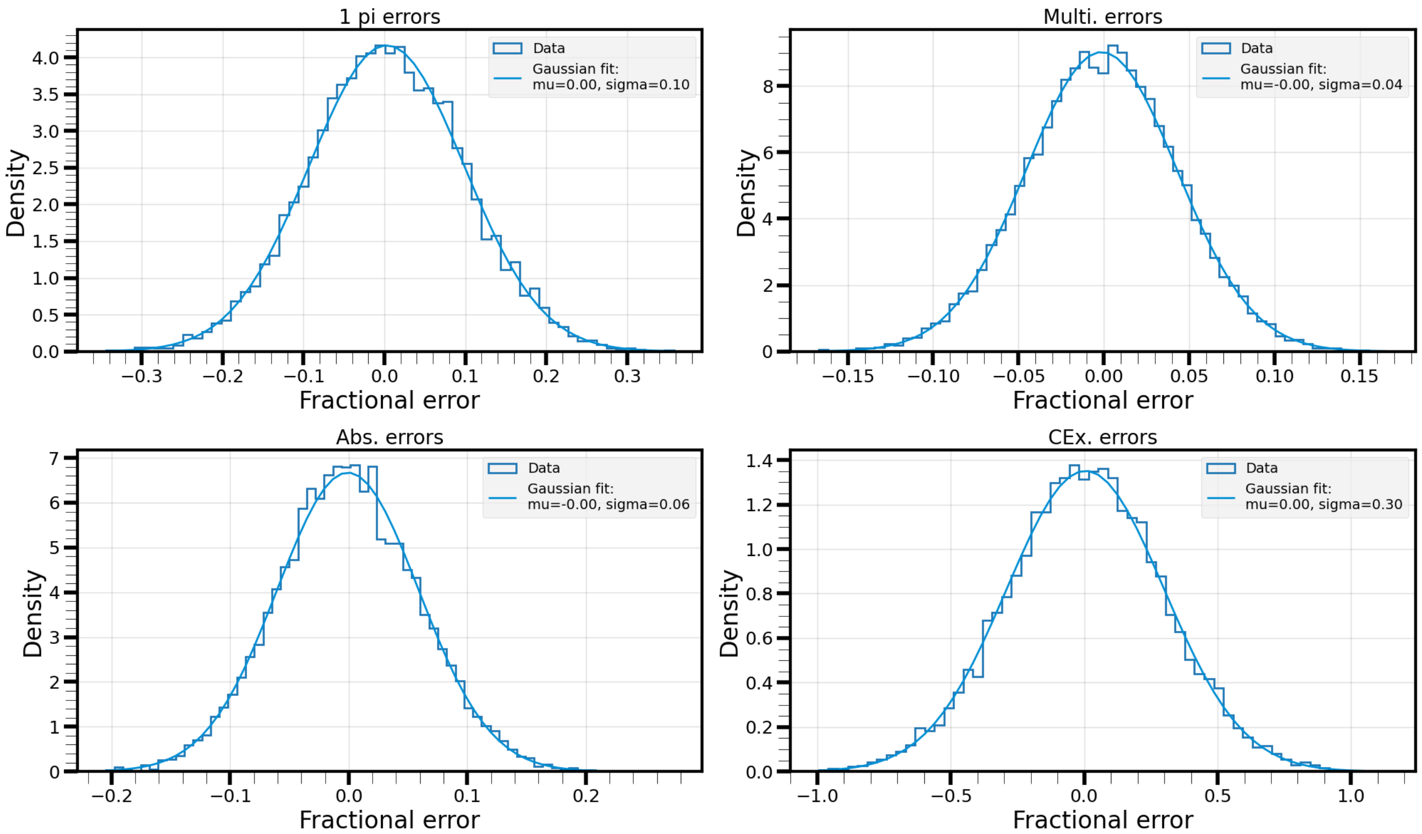
- Combining all MC events, and randomly splitting these removes pulls.



# “Mixing” fractional errors

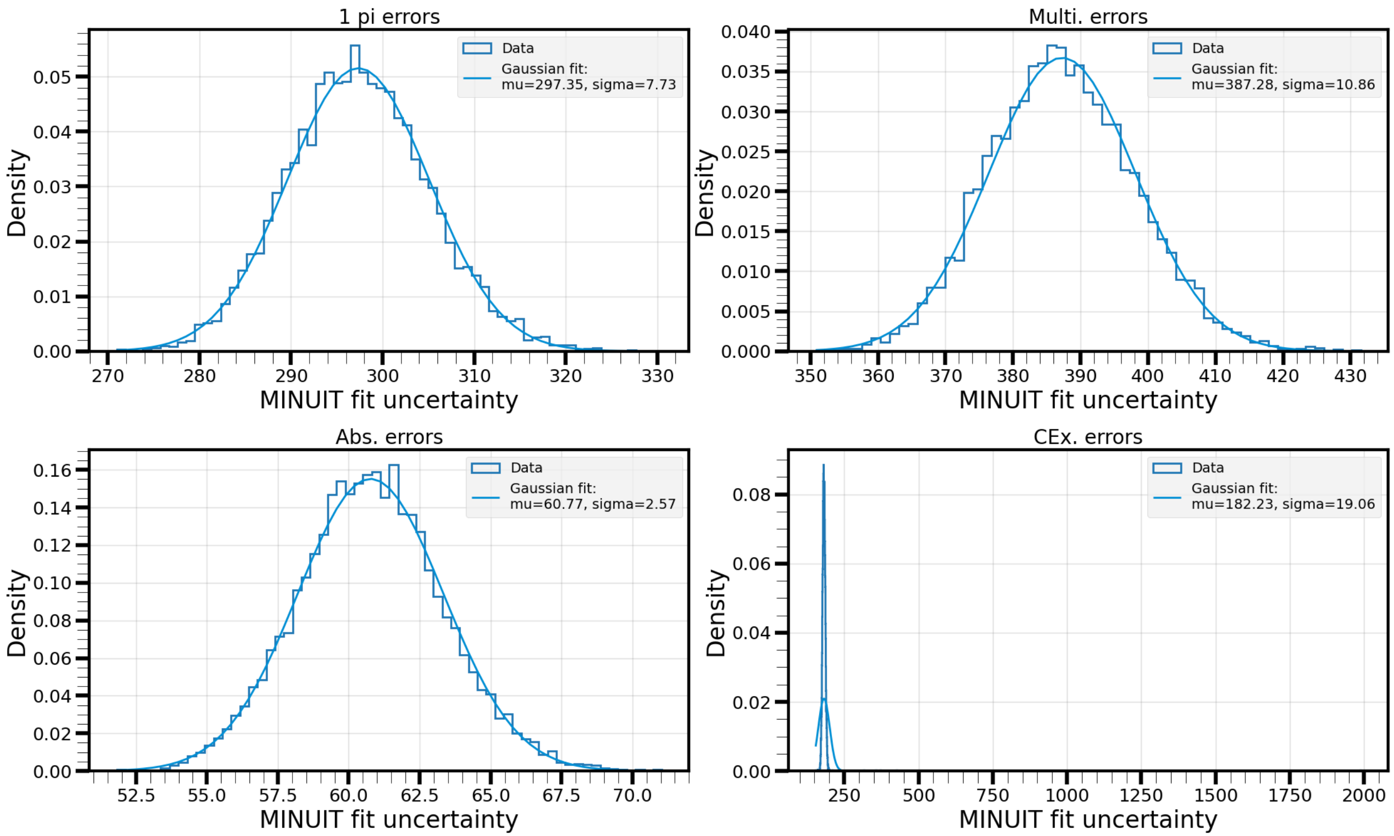
$$\frac{\text{Prediction}}{\text{True}} - 1$$

- Combining all MC events, and randomly splitting these removes pulls.



# “Mixing” fit uncertainties

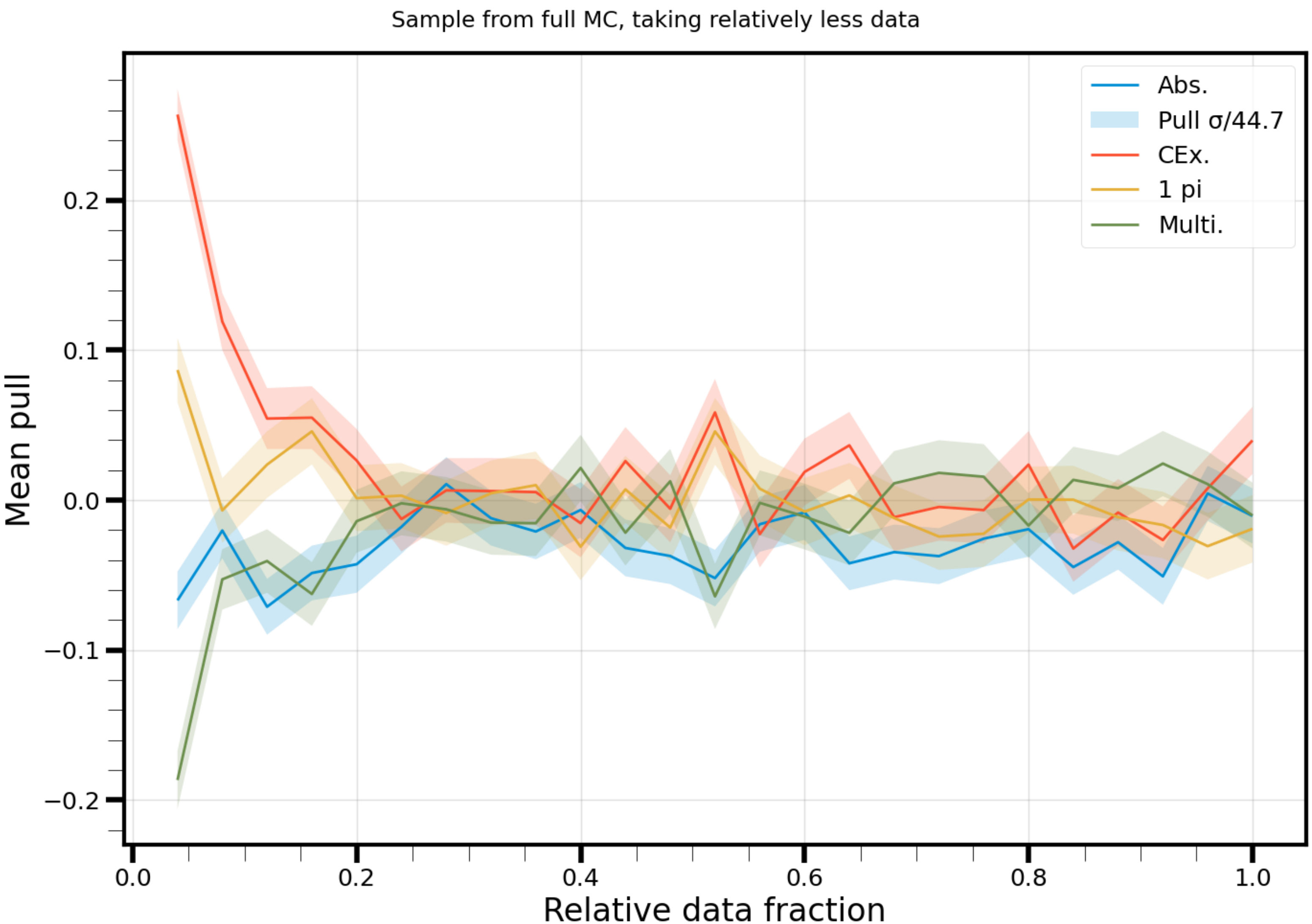
- Combining all MC events, and randomly splitting these removes pulls.



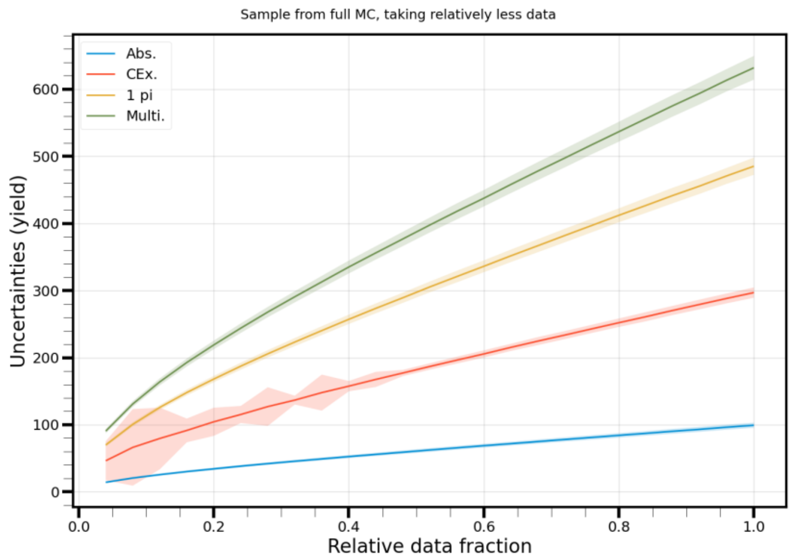
# Tests

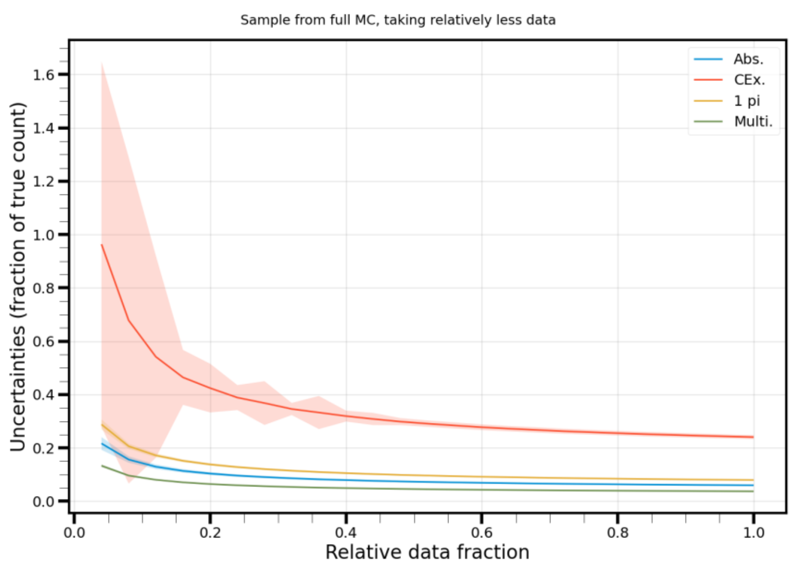
- Random fluctuation
- Data statistics
- Template statistics
- Re-weighted process fractions
- Initial fit predictions
- GNN score drift – some form of smearing the underlying distributions
  - Note: need to confirm Minuit works with non integer templates (nuisances **Poisson** distribute template bins)
- Outliers

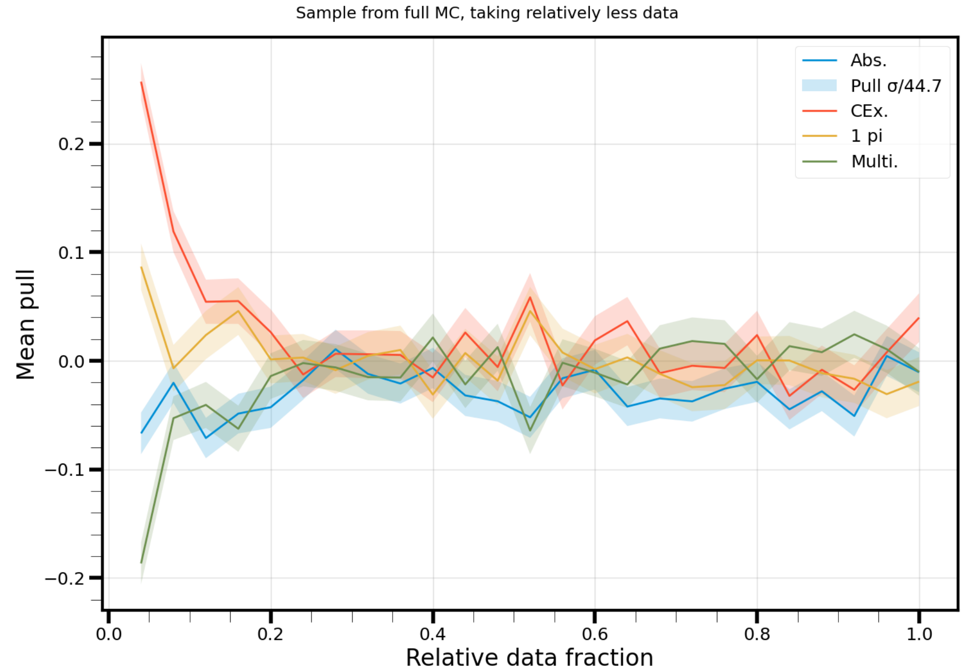
# Data statistics

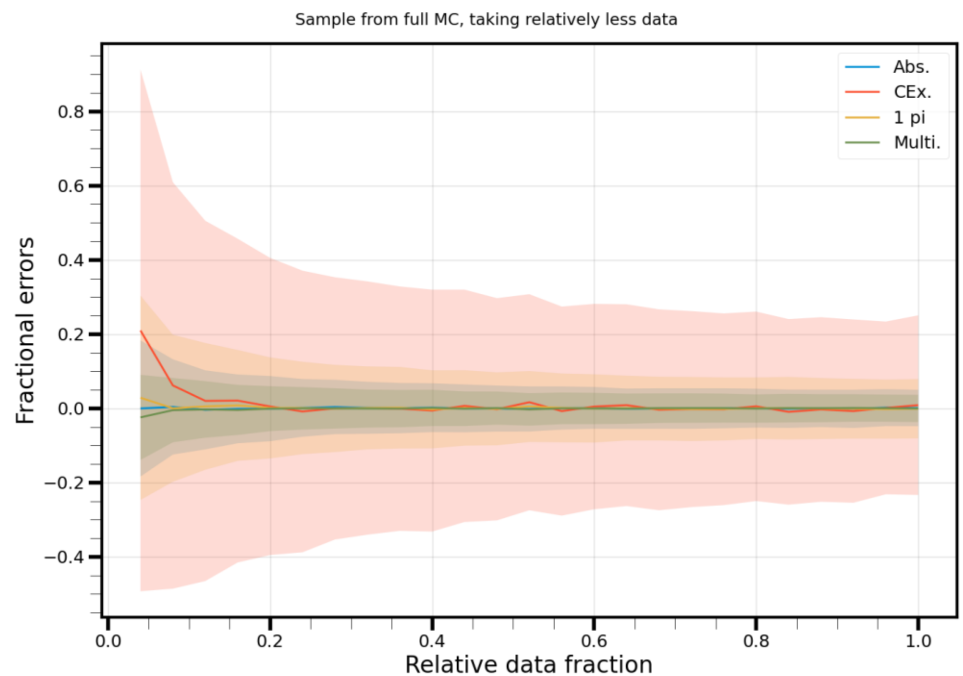


# Data statistics

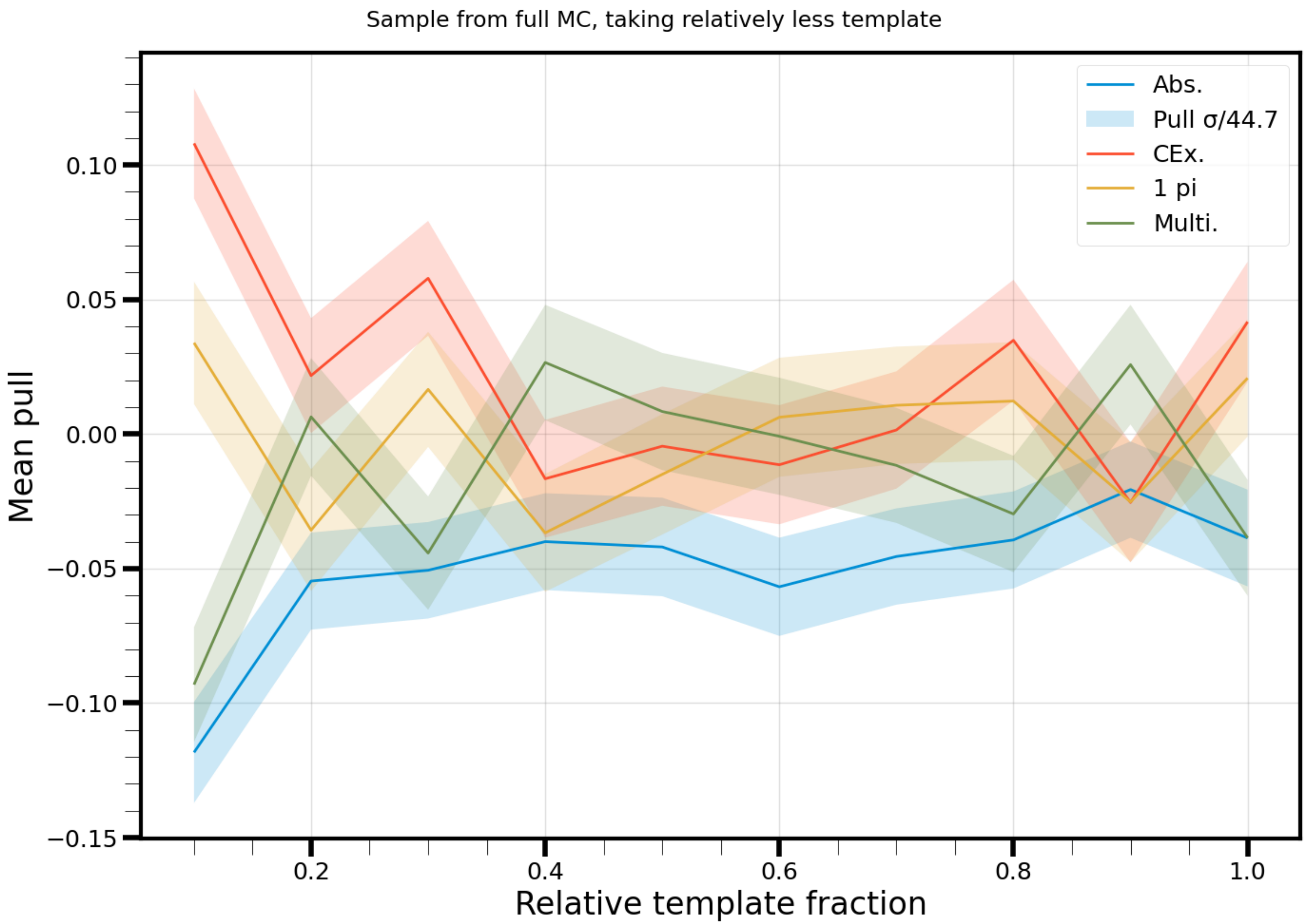


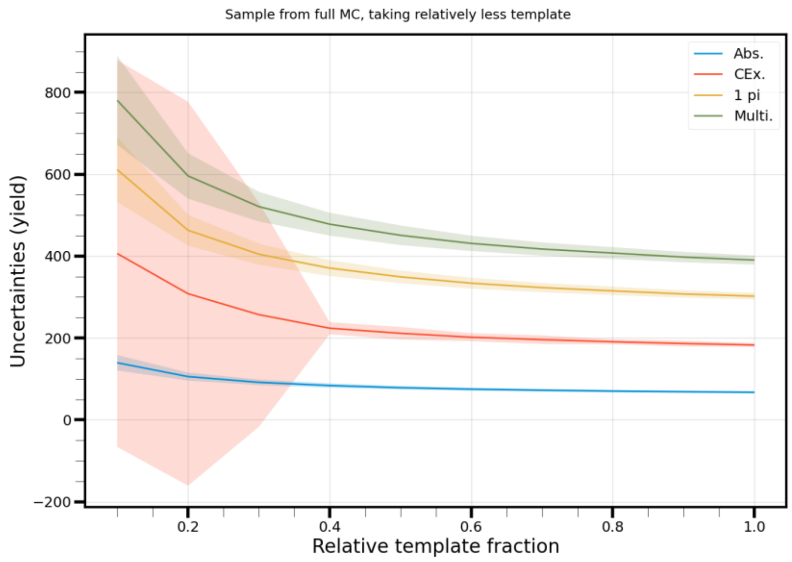



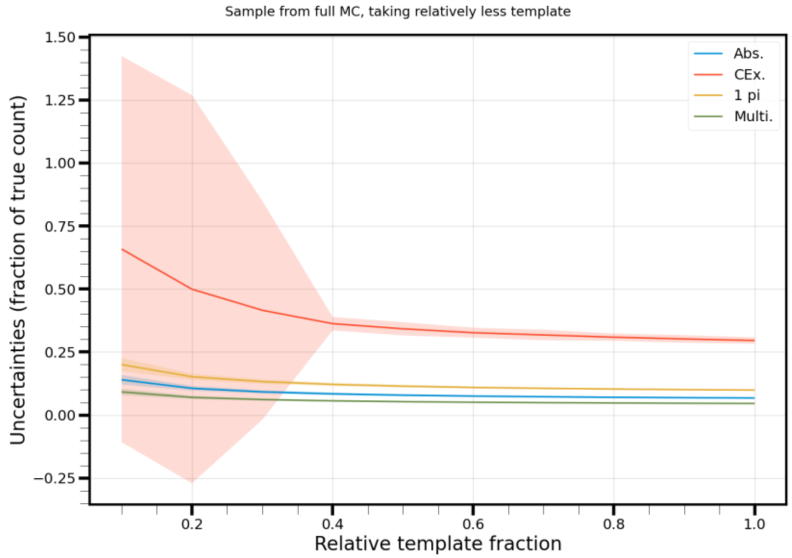



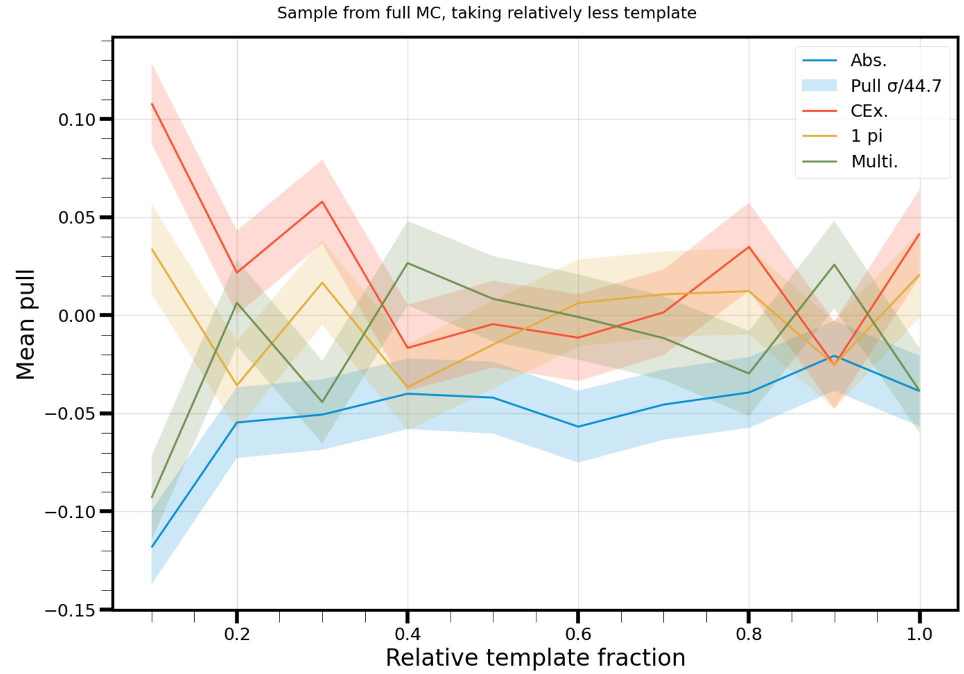
# Template statistics

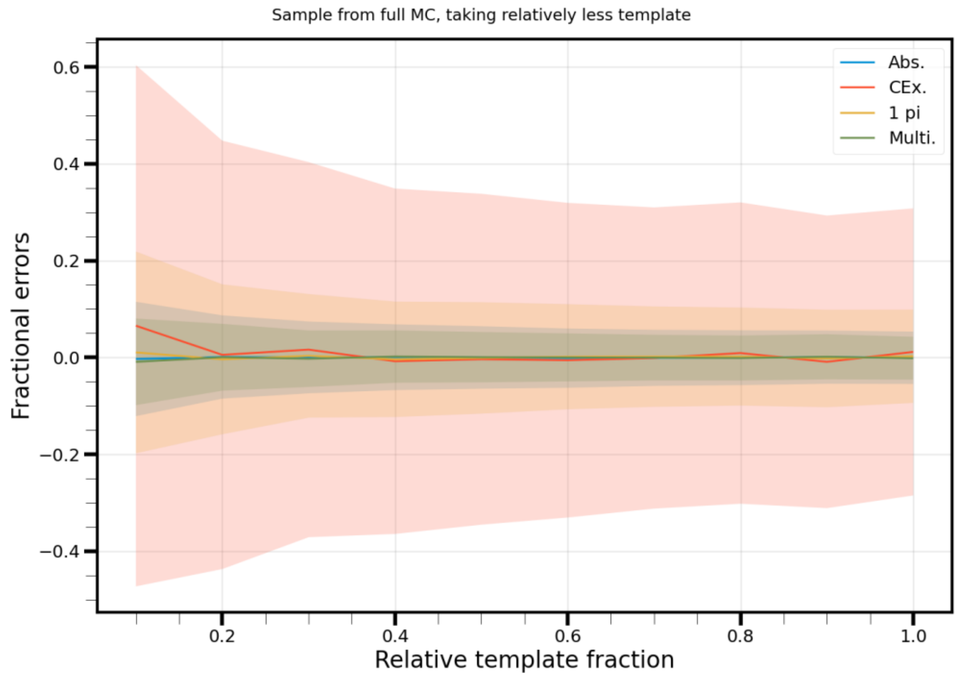


# Temp. statistics



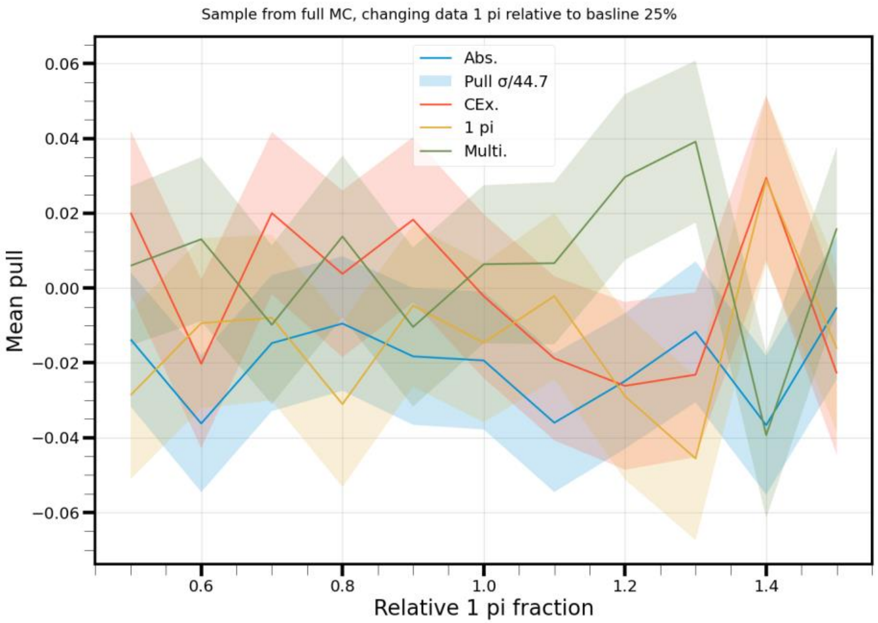



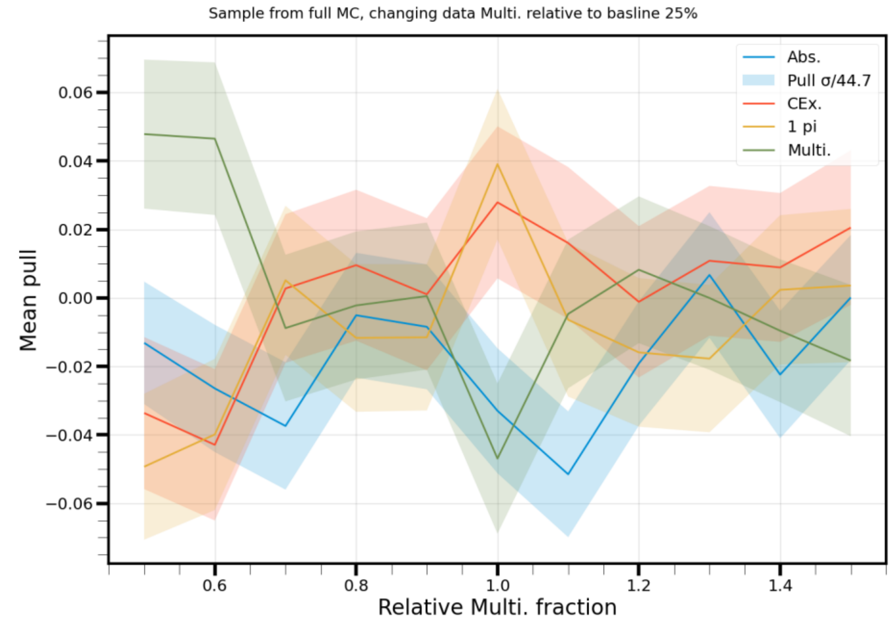



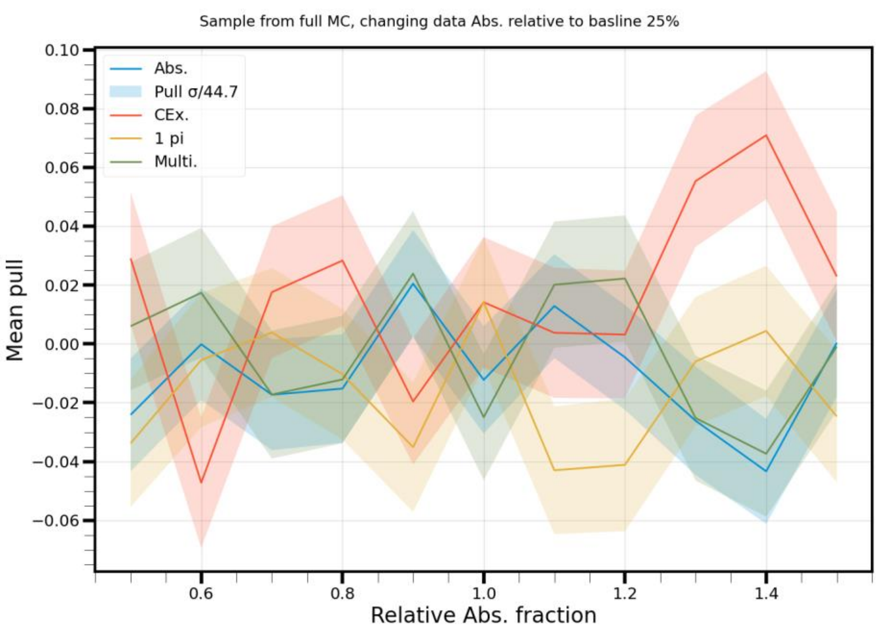



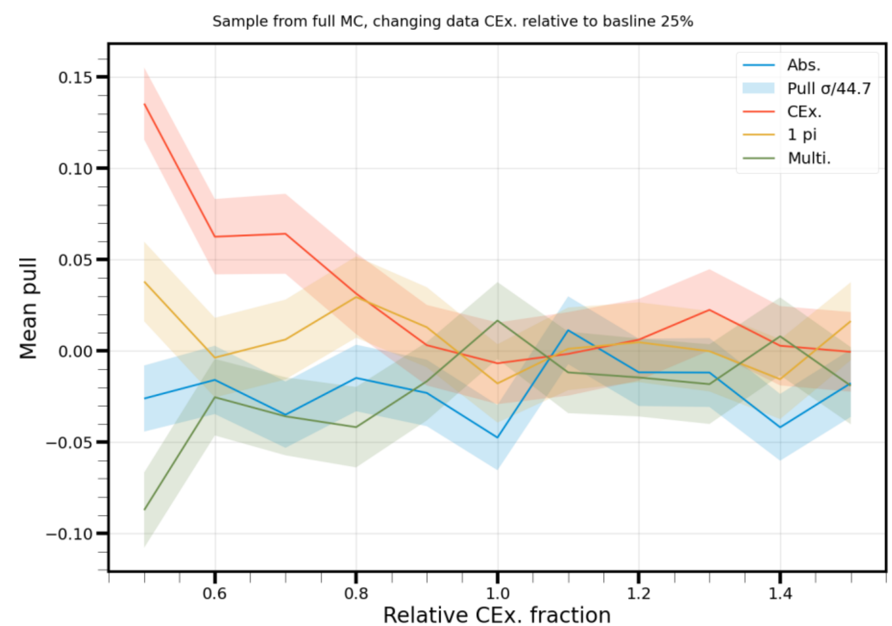
# Weighting data

A bad method should show a negative gradient (over-estimate when data has a deficit)



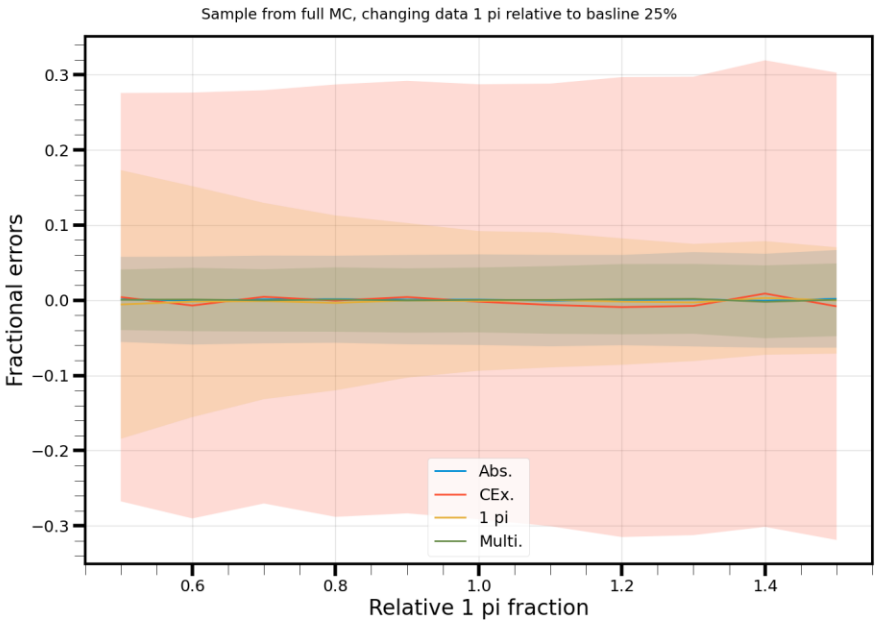



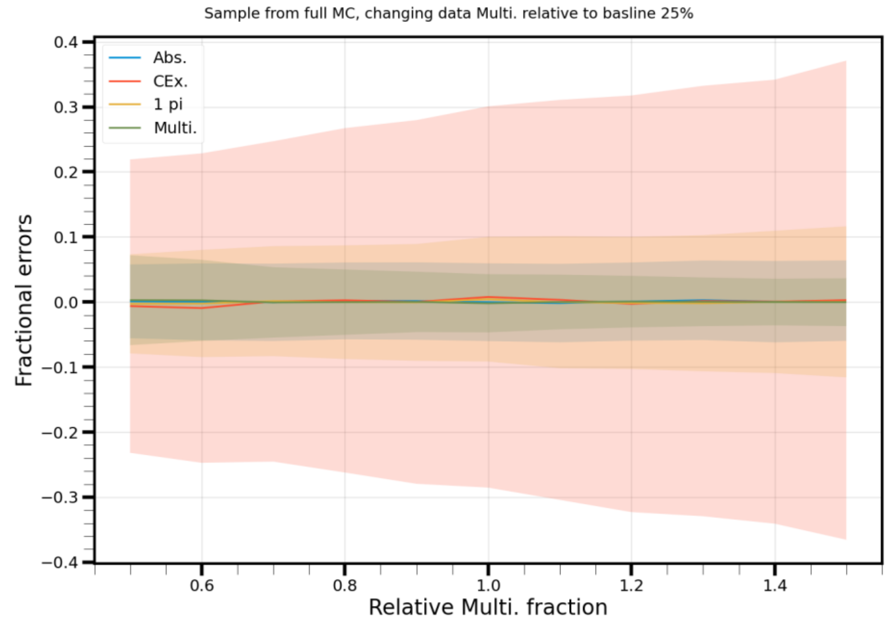



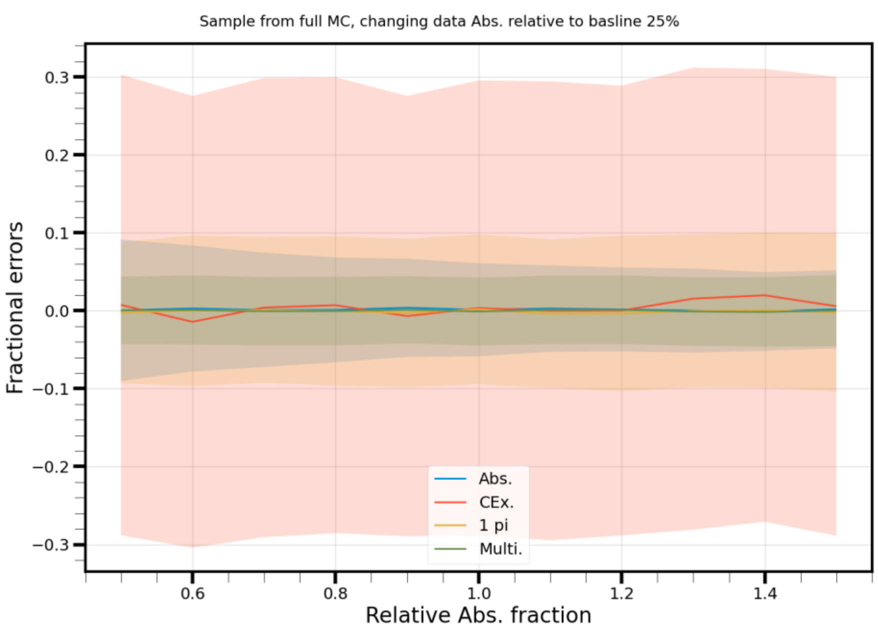



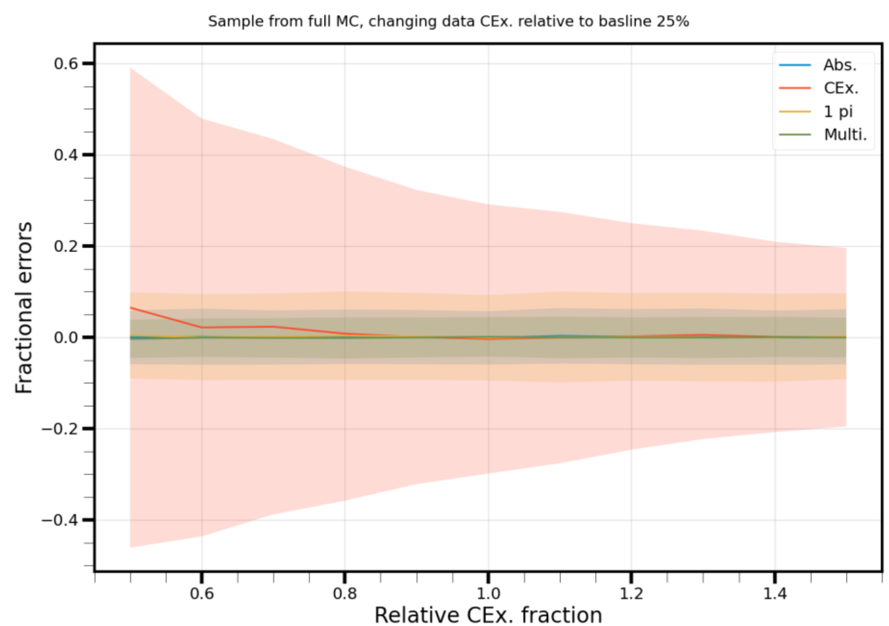
# Weighting data

A bad method should show a negative gradient (over-estimate when data has a deficit)



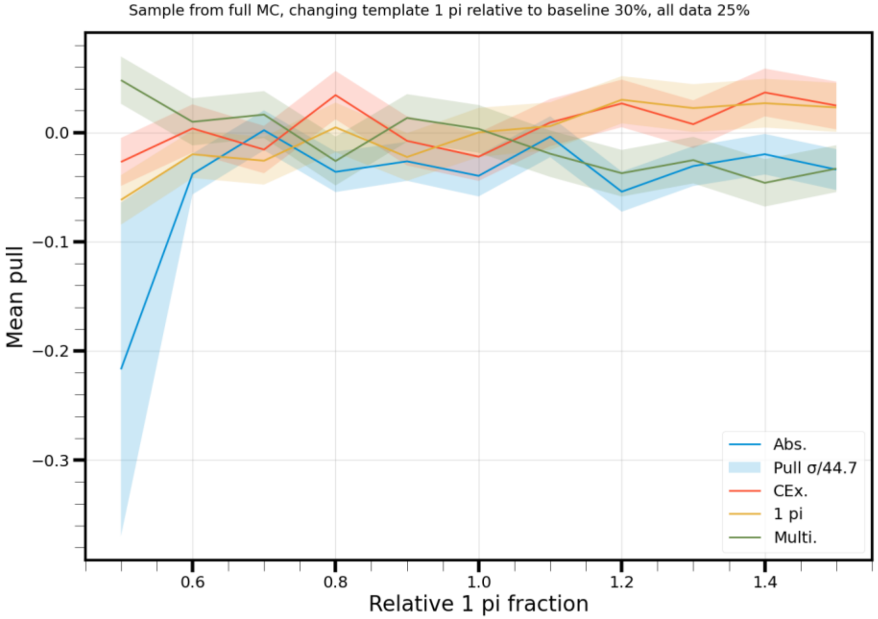



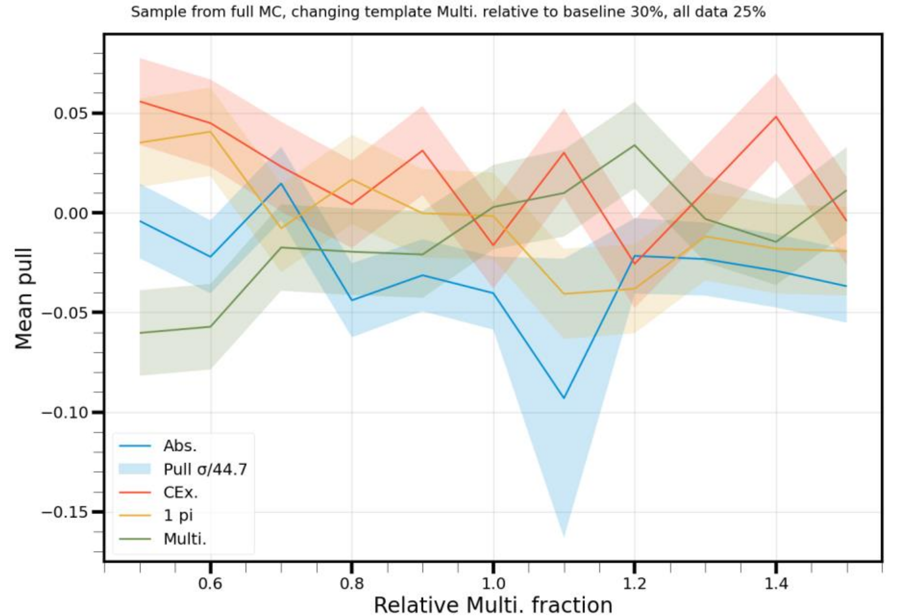



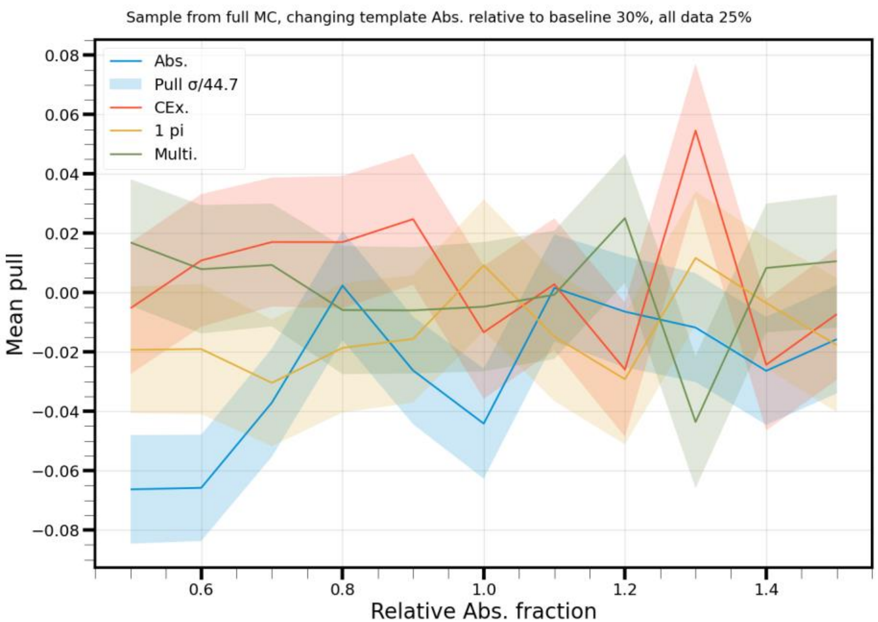



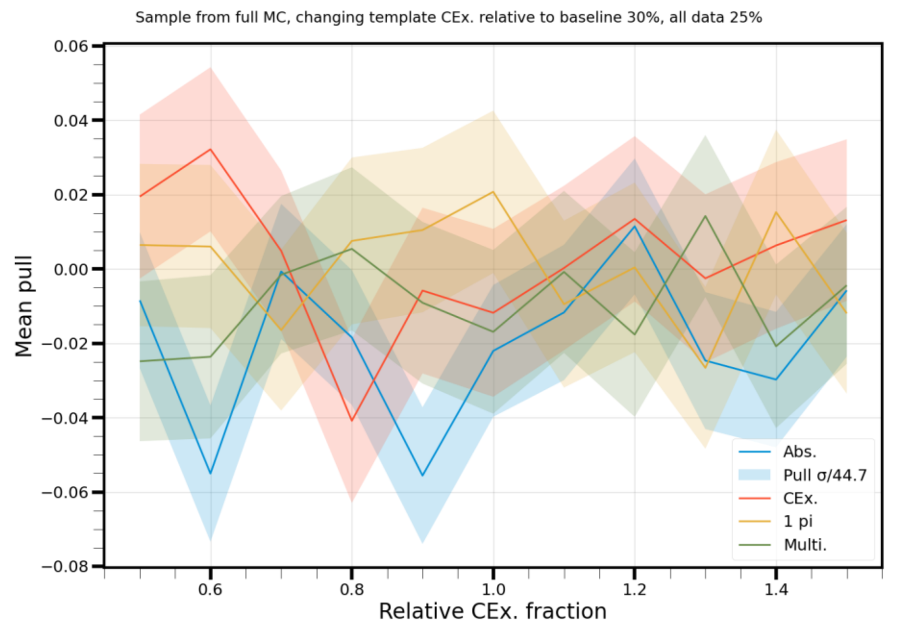
# Weighting template

A bad method should show a positive gradient
(under-estimate when template has a deficit)



# Weighting template

A bad method should show a positive gradient (under-estimate when template has a deficit)